View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Repositorio da Producao Cientifica e Intelectual da Unicamp

Universidade Estadual de Campinas

Instituto de Computação

Eduardo Moraes de Morais

CCA1-secure somewhat homomorphic encryption

Encriptação parcialmente homomórfica CCA1-segura

CAMPINAS

2016

Eduardo Moraes de Morais

CCA1-secure somewhat homomorphic encryption

Encriptação parcialmente homomórfica CCA1-segura

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Dissertation presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Supervisor/Orientador: Prof. Dr. Ricardo Dahab
Co-supervisor/Coorientador: Prof. Dr. Diego de Freitas Aranha

Este exemplar corresponde à versão final da Tese defendida por Eduardo Moraes de Morais e orientada pelo Prof. Dr. Ricardo Dahab.

CAMPINAS

2016

Agência(s) de fomento e nº(s) de processo(s): CNPq, 143484/2011-7; CAPES

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Maria Fabiana Bezerra Muller - CRB 8/6162

M792c

Morais, Eduardo Moraes de, 1983-
CCA1-secure somewhat homomorphic encryption / Eduardo Moraes de Morais. – Campinas, SP : [s.n.], 2016.

Orientador: Ricardo Dahab.
Coorientador: Diego de Freitas Aranha.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Criptografia homomórfica. 2. Computação verificável. 3. Ataques de texto cifrado escolhido (ATCE). I. Dahab, Ricardo,1957-. II. Aranha, Diego de Freitas,1982-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Encriptação parcialmente homomórfica CCA1-segura
Palavras-chave em inglês:
Homomorphic encryption
Verifiable computation
Chosen-ciphertext attacks (CCA)
Área de concentração: Ciência da Computação
Titulação: Doutor em Ciência da Computação
Banca examinadora:
Ricardo Dahab [Orientador]
Anderson Clayton Alves Nascimento
Alejandro Hevia Angulo
Julio César López Hernández
Antonio Carlos de Andrade Campello Junior
Data de defesa: 14-06-2016
Programa de Pós-Graduação: Ciência da Computação

Universidade Estadual de Campinas

Instituto de Computação

Eduardo Moraes de Morais

CCA1-secure somewhat homomorphic encryption

Encriptação parcialmente homomórfica CCA1-segura

Banca Examinadora:

- Ricardo Dahab
  Instituto de Computação, Universidade Estadual de Campinas
- Alejandro Hevia Angulo
  Department of Computer Science, Universidad de Chile
- Anderson Clayton Alves Nascimento
  Departamento de Engenharia Elétrica, Universidade de Brasília
- Julio César López Hernández
  Instituto de Computação, Universidade Estadual de Campinas
- Antonio Carlos de Andrade Campello Junior
  Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas

A ata da defesa com as respectivas assinaturas dos membros da banca encontra-se no processo de vida acadêmica do aluno.

Campinas, 14 de junho de 2016

Agradecimentos

Eu gostaria de agradecer à minha família, pelo apoio durante o programa de doutoramento; ao meu orientador Ricardo Dahab e ao meu co-orientador Diego Aranha, pelo suporte acadêmico de qualidade; aos funcionários, à Unicamp e ao Instituto de Computação, pela infraestrutura oferecida.

Eu gostaria de agradecer ao professor Steven Galbraith e ao Departamento de Matemática da Universidade de Auckland pelo apoio durante o período de 6 meses que permaneci lá pelo programa Ciências sem Fronteiras.

Resumo

Nesta tese nosso tema de pesquisa é a encriptação homomórfica, com foco em uma solução prática e segura para encriptação parcialmente homomórfica (*somewhat homomorphic encryption* - SHE), considerando o modelo de segurança conhecido como ***ataque de texto encriptado escolhido*** (*chosen ciphertext attack* - CCA). Este modelo pode ser subdividido em duas categorias, a saber, CCA1 e CCA2, sendo CCA2 o mais forte. Sabe-se que é impossível construir métodos de encriptação homomórfica que sejam CCA2-seguros. Por outro lado, é possível obter segurança CCA1, mas apenas um esquema foi proposto até hoje na literatura [65]; assim, seria interessante haver outras construções oferecendo este tipo de segurança.

Resumimos os principais resultados desta tese de doutorado em duas contribuições. A primeira é mostrar que a família NTRU de esquemas SHE é vulnerável a ataques de recuperação de chave privada, e portanto não são CCA1-seguros. A segunda é a utilização de computação verificável para obter esquemas SHE que são CCA1-seguros e que podem ser usados para avaliar polinômios multivariáveis quadráticos.

Atualmente, métodos de encriptação homomórfica são construídos usando como substrato dois problemas de difícil solução: o MDC aproximado (*approximate GCD problem* - AGCD) e o problema de aprendizado com erros (*learning with errors* - LWE). O problema AGCD leva, em geral, a construções mais simples mas com desempenho inferior, enquanto que os esquemas baseados no problema LWE correspondem ao estado da arte nesta área de pesquisa. Recentemente, Cheon e Stehlé [29] demonstraram que ambos problemas estão relacionados, e é uma questão interessante investigar se esquemas baseados no problema AGCD podem ser tão eficientes quanto esquemas baseados no problema LWE. Nós respondemos afirmativamente a esta questão para um cenário específico: estendemos o esquema de computação verificável proposto por

Fiore, Gennaro e Pastro [39], de forma que use a suposição de que o problema AGCD é difícil, juntamente com o esquema DGHV adaptado para uso do Teorema Chinês dos Restos [28] (*Chinese remainder theorem* - CRT) de forma a evitar ataques de recuperação de chave privada.

Abstract

In this thesis we study homomorphic encryption with focus on practical and secure somewhat homomorphic encryption (SHE), under the ***chosen ciphertext attack*** (CCA) security model. This model is classified into two different main categories: CCA1 and CCA2, with CCA2 being the strongest. It is known that it is impossible to construct CCA2-secure homomorphic encryption schemes. On the other hand, CCA1-security is possible, but only one scheme is known to achieve it [65]. It would thus be interesting to have other CCA1-secure constructions.

The main results of this thesis are summarized in two contributions. The first is to show that the NTRU-family of SHE schemes is vulnerable to key recovery attacks, hence not CCA1-secure. The second is the utilization of verifiable computation to obtain a CCA1-secure SHE scheme that can be used to evaluate quadratic multivariate polynomials.

Homomorphic encryption schemes are usually constructed under the assumption that two distinct problems are hard, namely the ***Approximate GCD*** (AGCD) Problem and the ***Learning with Errors*** (LWE) Problem. The AGCD problem leads, in general, to simpler constructions, but with worse performance, wheras LWE-based schemes correspond to the state-of-the-art in this research area. Recently, Cheon and Stehlé [29] proved that both problems are related, and thus it is an interesting problem to investigate if AGCD-based SHE schemes can be made as efficient as their LWE counterparts. We answer this question positively for a specific scenario, extending the verifiable computation scheme proposed by Fiore, Gennaro and Pastro [39] to work under the AGCD assumption, and using it together with the Chinese Remainder Theorem (CRT)-version of the DGHV scheme [28], in order to avoid key recovery attacks.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

> The scientist is not a person who gives the right answers, he's one who asks the right questions.
>
> Claude Lévi-Strauss

The main body of work of this thesis is presented in the form of a collection of publications. In order to ease the burden on the reader, our goal in this chapter is to give a summary of the main results and show how they are organized along the research process.

We proceed by describing interesting problems in homomorhic encryption and providing arguments that justify the choice of certain solutions we made in the thesis. Each target problem will be given in the form of an interesting question that must receive a satisfying answer. We also present the definitions and concepts necessary to understand why the proposed questions are interesting, and explain how to measure the quality of the obtained solutions.

1.1 Motivation

It is common practice to analyze cryptographic solutions based on two main goals: ***security*** and ***efficiency***. Usually, the analysis consists in understanding the tradeoffs between these properties, according to the objectives imposed by the conditions and circumstances of a specific scenario. Furthermore, we are going to consider another property that in many cases is ignored in Cryptography, that we will call ***functionality***. Although functionality seems in principle to be orthogonal to security, since an encrypted message in general can not be meaningfully manipulated, we are going to see that in fact it is possible to offer some kind of functionality to a cryptosystem.

Our work focuses on ***lattice-based cryptography***. More specifically, we study ***homomorphic encryption***, which, before Gentry's breakthrough in 2009, when he proposed the first ***fully homomorphic encryption*** (FHE) construction [42, 43], was re-

garded as Cryptography's holy grail, since it allows, in some sense, maximal functionality. Here we measure functionality as a kind of ciphertext flexibility. Namely, suppose we want to compute an algorithm f over some private data x. Using FHE we can compute ***any*** such algorithm using as input the encryption of x and producing as output the encryption of the desired result, i.e., the encryption of $f(x)$. On the other hand, although FHE offers maximum flexibility, we have that, for a reasonable security level, the efficiency of existent FHE proposals is far from practical.

Ideally, one would choose to maximize security, efficiency and functionality. However, this perfect solution is far from achievable. Thus, we begin by relaxing our expectations, and describing what we accept as a good approximation of our ideal goals. We accomplish this task taking into consideration a real world scenario, where cryptography must be efficient enough to be used in practice. Then, we will not be interested so much in solutions whose performance is infeasible.

We also want to be able to have some functionality over our encrypted data. Here, we accept the restriction of working with useful functions that are necessary to solve some particular problem. For example, homomorphic encryption has being used to solve problems in the health and financial areas [15, 16, 59, 79]. Among functions that are important in these scenarios, we have the computation of statistical functions, linear regression, searching and sorting encrypted data, edit distance, and many others. But FHE is too powerful for such functions, which can be evaluated by simply using a more efficient and less flexible construction called ***somewhat homomorphic encryption*** (SHE) [21]. Then, instead of maximal functionality, we accept a restricted flexibity in the ability to compute over encrypted data that is good enough to solve real-world problems. Finally, we consider that a cryptosystem is secure if it is appropriate for utilization in cloud computing. Since almost every SHE proposal is succeptible to ***key recovery attacks*** [32], and because such kind of attacks is a real threat in cloud computing, then our goal here is to obtain a construction that is secure against such kind of attacks.

Now that we have informally relaxed our goals, we present the high-level version of the interesting question this PhD thesis must answer:

> **How can we construct practical and useful SHE schemes that resist key-recovery attacks?**

In what follows we formalize the high-level notion stated above.

1.2 Thesis organization

In this section we summarize the problems and results obtained in this PhD thesis, pointing out in which chapter the subject is treated and the underlying publications corresponding to each contribution.

During the PhD program I have participated in a research project about efficient implementation of elliptic curve protocols for Android architecture, giving rise to two publications [17, 18]. The research consisted in bridging JNI and JCA interfaces to interact with the Relic library [6] and a set of cryptographic primitives like for example ECDSA, ECSTS, ECDH, ECIES, short signatures, Salsa20, Blake and Serpent. Different security levels were considered in order to obtain a time comparison of the primitives. Since this research area is not related to homomorphic encryption, we are not going to explore it further here.

In Chapter 1, we introduce the reader to the subject. In particular, we give fundamental tools which are necessary to understand the next chapters, with an initial focus on abstract algebra theory, because we feel that this material can be useful for other students that may have a good background on algorithms and programming, but need to improve the mathematical knowledge that is required in the construction of homomorphic encryption schemes. We shortly describe some important definitions and theorems that are important for lattice-based cryptography and present some crptographic primitives.

In Chapter 2, we put together the publications that were the result of the research conducted during the PhD program. First, we mention our contribution to a book chapter, that corresponds to an introduction to Lattice-based Cryptography [10]. This work derived from a short course in 2013 [9] and gave rise to a technical report [78]. Following that, we present an introduction to homomorphic encryption [33], which is the text of a short course that gave rise to another technical report [77].

The next paper is a key-recovery attack to a family of homomorphic encryption schemes. The ability to compute private keys using decryption oracles is a serious problem in cloud computing, because the cloud can monitor the client behaviour in order to test the validity of ciphertexts, by submitting queries and testing whether the client returns an error message or not. Formally, a cryptosystem must be shown to be secure in the CCA1 model in order to avoid key-recovery attacks. Unfortunately, there is only one SHE scheme that is CCA1-secure [65]. One of the main results of this thesis, presented in the third paper of Chapter 2, is a key-recovery attack to the NTRU-based family of SHE schemes [32]. The paper remarks that it would be interesting to have more proposals achieving CCA1-security.

Finally, we show in the last paper of Chapter 2 how verifiable computation can be used to construct CCA1-secure SHE schemes [76], for the case of quadratic multivariate polynomial functions. We have adapted the work of Fiore, Gennaro and Pastro [39] to use the AGCD problem, instead of the LWE problem, using the Chinese Remainder Theorem (CRT) to encode more information inside plaintexts, which allowed us to obtain a construction with lower overhead. In this paper we also answer, positively, the question as to whether AGCD-based Cryptography can be as efficient as LWE-based Cryptography, for the same security level. Precisely for this restricted class of quadratic multivariate polynomials, the AGCD-based construction presents interesting efficiency characteristics [76]. Using the AGCD-based SHE scheme, it is possible

to homomorphically compute one multiplication by choosing roughly $\eta = 2\rho$. Moreover, we can use the CRT to encode ℓ plaintext slots inside each ciphertext, for large ℓ. Asymptotically, we have that the plaintext size can be made as big as half the size of the ciphertext, a condition that cannot be achieved using the LWE-based scheme.

In Chapter 3 we discuss how the papers relate to each other. We analyze practical instantiations of AGCD-based and LWE-based constructions, based on the best-attack running time.

In Chapter 4 we conclude the thesis and give final remarks. We also point out future directions of investigation.

1.3 Abstract Algebra

This section provides basic abstract algebra definitions and theorems. We also define homomorphisms, which is a central figure not only in the study of algebra, but also in lattice-based cryptography. For further information on this subject, we point the reader to Dummit and Foot's book [36].

1.3.1 Groups

Definition 1.3.1. A ***group*** is defined by the pair $(G, \circ)$, where $\circ$ is a ***closed associative binary operation*** over a set G, that contains an ***identity element***. Furthermore, every element from G has ***inverse*** in G. If the group respects the ***commutative property*** it is called ***Abelian*** group.

Example 1.3.1. The set of integer numbers $\mathbb{Z}$ and the addition operation forms an Abelian group, whose identity element is 0. Each element $a \in \mathbb{Z}$ has inverse given by $-a \in \mathbb{Z}$. The set of non-zero real numbers $\mathbb{R}$ with usual multiplication forms an Abelian group, whose identity is 1. Each element $a \in \mathbb{R}$ has inverse given by $1/a \in \mathbb{R}$. Let $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$. Given $a, b \in \mathbb{Z}_n$, we define the operations $\oplus$ and $\odot$ as follows:

$$\begin{aligned} a \oplus b &= a + b \pmod{n} \\ a \odot b &= a.b \pmod{n}. \end{aligned}$$

Thus, $(\mathbb{Z}_n, \oplus)$ forms an Abelian group, whose identity is again 0. Each element $a \in \mathbb{Z}_n$ has inverse given by $(n - a) \in \mathbb{Z}_n$. Moreover, let $\mathbb{Z}_n^* = \{1, \ldots, n-1\}$. Then $(\mathbb{Z}_n^*, \odot)$ forms a Abelian group if and only if n is a prime number. The multiplicative inverse $a \in \mathbb{Z}_n^*$ can be computed solving the Diophantine equation $a.x + n.y = 1$. As this equation has only one solution modulo n if $\mathrm{GCD}(a, n) = 1$, we have that, for a non-prime n, the elements $a \in \mathbb{Z}_n^*$ that have no inverses are such that $\mathrm{GCD}(a, n) \neq 1$. As long as there is no ambiguity, the symbols $+$ and $.$ are used instead of $\oplus$ and $\odot$, respectively.

Example 1.3.2. Let $\mathbb{Z}$ be the group defined in example 1.3.1 and $2\mathbb{Z} = \{\dots, -4, -2, 0, 2, 4, \dots\}$, formed by the addition of each element of $\mathbb{Z}$ with itself, obtaining the set of even numbers. Then $2\mathbb{Z}$ is the subgroup of $\mathbb{Z}$, because given two elements $a, b \in 2\mathbb{Z}$, $a + b$ is an even number, hence it belongs to $\mathbb{Z}$. Moreover, given an element of $a \in 2\mathbb{Z}$, there is $-a \in 2\mathbb{Z}$, such that $-a$ is the inverse of a.

Quotient group

Definition 1.3.2. Given a group $(G, \circ)$, a subgroup H and $a \in G$, we define the ***left coset*** as the set given by $a \circ H = \{a \circ h \mid h \in H\}$. Analogously, we define the ***right coset*** as the set given by $H \circ a = \{h \circ a \mid h \in H\}$.

If G is an Abelian group, then $a \circ H = H \circ a$ and hence there is no difference between left cosets and right cosets.

Definition 1.3.3. Given a finite group G, we define the ***order*** of G by the number of elements in the set G.

Theorem 1.3.1. Let H be a subgroup of G, where G is a finite group. Then the order of H divides the order of G.

Proof. It is easy to see that if $a \notin H$, then $a \circ H \cap H = \{\oslash\}$, because otherwise we would have $a \circ h_1 = h_2$, for $h_1, h_2 \in H$. Thus $a = h_2 \circ h_1^{-1}$ and hence $a \in H$, establishing a contradiction. Moreover, $|a \circ H| = |H|$, because otherwise we would have distinct elements $h_1, h_2 \in H$, such that $a \circ h_1 = a \circ h_2$, but this implies that $a^{-1} \circ a \circ h_1 = a^{-1} \circ a \circ h_2$ and then we have that $h_1 = h_2$, a contradiction. Thus, G can be partitioned in cosets derived from H. Namely, $G = H \cup a_1 \circ H \cup \dots \cup a_k \circ H$, where a_i does not belong to H and also does not belong to no other coset $a_j \circ H$, para $i \neq j$.

Therefore, $|G| = (k + 1).|H|$. □

Definition 1.3.4. The ***quotient group*** G/H is defined as being formed by equivalence classes generated by the partitioning of G with respect to H. If this partitioning is given by $\{H, a_1 \circ H, \dots, a_k \circ H\}$, the operation $\star$ over the quotient group is defined by

$$(a_i \circ H) \star (a_j \circ H) = (a_i \circ a_j) \circ H.$$

Corolary 1.3.1. For a finite group G and some subgroup $H \subset G$, we have that $|G/H| = |G|/|H|$.

For example, the quotient group $\mathbb{Z}/n\mathbb{Z}$, denoted also by $\mathbb{Z}_n$, is formed by the cosets:

$$\begin{aligned} n\mathbb{Z} &= \{0, n, 2n, \dots\}, \\ n\mathbb{Z}+1 &= \{1, n+1, 2n+1, \dots\}, \\ &\vdots \\ n\mathbb{Z}+(n-1) &= \{n-1, 2n-1, 3n-1, \dots\}. \end{aligned}$$

Homomorphisms

Definition 1.3.5. The function $f : G \to H$ is called a ***homomorphism*** from G to H, if f preserves operations of group G. In other words, if $\circ$ and $\star$ are the operations of G and H respectively, then we say that f preserves the operation of G if for any $a, b \in G$, then $f(a \circ b) = f(a) \star f(b)$. If, in addition, f is a bijection, then f is denominated ***isomorphism***. If f is a bijection from G to G, then f is called ***automorphism***.

Theorem 1.3.2. Let $f : G \to H$ be a homomorphism between groups G and H. If $e \in G$ represents the identity element of G, then $f(e)$ represents the identity of H.

Proof. Using the definition of homomorphisms together with the fact that $e.e = e$, we have $f(e).f(e) = f(e)$. Therefore, $f(e)$ is the identity element of H. □

Theorem 1.3.3. Let $f : G \to H$ be a homomorphism between the groups G and H. Then f maps inverses from G to inverses of H. In other words, for every $a \in G$, we have that $f(a^{-1}) = (f(a))^{-1}$.

Proof. For any $a \in G$, we have that $a.a^{-1} = e$. Thus, $f(a)f(a^{-1}) = f(e)$. Hence, $f(a^{-1}) = (f(a))^{-1}$. □

Example 1.3.3. An important example of automorphism of a group G, called ***inner automorphism***, is the provided by conjugation by a fixed element in G. Namely, $f_a : G \to G$, such that, for any $a \in G$, then $f_a(x) = axa^{-1}$. Then elements x and axa^{-1} are called ***conjugates***. Given a subgroup S of G, the set $aSa^{-1} = \{asa^{-1} \mid s \in S\}$, for $a \in G$, is denominated ***conjugate*** of a subgroup S.

Let $f : G \to H$ be a homomorphism from G to H. The set $N = \{a \mid f(a) = e'\}$, where e' represents the identity element of H, is called the ***kernel*** of f, denoted by $Ker(f)$.

Theorem 1.3.4. Let $f : G \to H$ be a homomorphism from G to H. Then $Ker(f)$ is a subgroup of G.

Proof. We have that $\mathrm{Ker}(f)$ is a subgroup of G, because for every pair of elements $a, b \in Ker(f)$, we have that $f(a) = e'$ and $f(b) = e'$, hence $f(a.b) = f(a).f(b) = e'.e' = e'$. Thus, $a.b$ belongs to $Ker(f)$. Furthermore, for any $a \in Ker(f)$, we have that $f(a) = e'$. Thus, $f(e) = f(a.a^{-1}) = f(a).f(a^{-1}) = e'$. Consequently, $e'.f(a^{-1}) = e'$. Therefore, $f(a^{-1}) = e'$, then $a^{-1} \in Ker(f)$. □

Let $f : \mathbb{Z} \to \mathbb{Z}_n$, given by $f(a) = a \pmod n$. The kernel $Ker(f)$ is formed by the integers a such that $a \equiv 0 \pmod n$. Thus, $\mathrm{Ker}(f)$ is formed by all multiples of n. Moreover, the set of multiples of n, denoted by $n\mathbb{Z}$, is a subgroup of $\mathbb{Z}$.

Definition 1.3.6. Let H be a subgroup of G, then H is called ***normal subgroup*** of G, if for every $h \in H$ and every $a \in G$, then $aha^{-1} \in H$.

Theorem 1.3.5. Let H be a subgroup of G. H is normal if and only if H is equal to its conjugates. Equivalently, H is normal if and only if H is invariant with respect to any inner automorphism in G.

Proof. If H is normal, according to the definition, we have that for $h \in H$ and $a \in G$, then $aha^{-1} \in H$. Thus, $aHa^{-1} \subset H$. To show that $H = aHa^{-1}$, for all $a \in G$, we must show that there is no pair of distinct elements $h_1, h_2 \in H$, such that $ah_1a^{-1} = ah_2a^{-1}$. Suppose by contradiction that there is such a pair of elements h_1 and h_2, multiplying on the right by a, we have that $ah_1 = ah_2$. Multiplying on the left by a^{-1}, we have that $h_1 = h_2$, a contradiction.

Moreover, if $H = aHa^{-1}$ for every $a \in G$, then for any $h \in H$ and for any $a \in G$, we have that $aha^{-1} \in H$, then H is normal. □

Theorem 1.3.6. Let H be a subgroup of G. H is normal if and only if every left coset aH is equal to the respective right coset Ha, for all $a \in G$.

Proof. If H is normal, by theorem 1.3.5 we have that $H = aHa^{-1}$, therefore $Ha = (aHa^{-1})a = aH$. □.

Theorem 1.3.7. Let $f : G \to H$ be a homomorphism form G to H. Then the kernel $\mathrm{Ker}(f)$ is a normal subgroup of G. Furthermore, H is isomorphic to the quotient group $G/\mathrm{Ker}(f)$. Reciprocally, if N is a normal subgroup of G, then the map $g : G \to G/N$, defined by $g(a) = aN$, for $a \in G$, is a homomorphism from G to G/N with kernel $\mathrm{Ker}(g) = N$.

Proof. By Theorem 1.3.4 we have that $\mathrm{Ker}(f)$ is a subgroup of G, because for every two elements $a, b \in \mathrm{Ker}(f)$, we have that $f(a) = f(b) = e'$, hence $f(a).f(b) = f(a.b)$. Thus, $a.b \in \mathrm{Ker}(f)$. Moreover, for every $a \in \mathrm{Ker}(f)$, we have that $f(e) = e'$, then $f(a.a^{-1}) = e'$ and then we obtain $f(a).f(a^{-1}) = e'$. Hence, $f(a^{-1}) = e'$ and thus we have that $a^{-1} \in \mathrm{Ker}(f)$. To show that it is a normal subgroup, it is enough to us the fact that for any $a \in \mathrm{Ker}(f)$, we have that $f(a) = e'$. Therefore, for every element $g \in G$, we have to show that $gag^{-1} \in \mathrm{Ker}(f)$. But applying the function f, we have that $f(gag^{-1}) = f(g).f(a).f(g)^{-1}$. As $f(a) = e'$ and $f(g^{-1}) = f(g)^{-1}$, we have that $f(gag^{-1}) = f(g).f(g)^{-1} = e'$. Therefore, $gag^{-1} \in \mathrm{Ker}(f)$. □

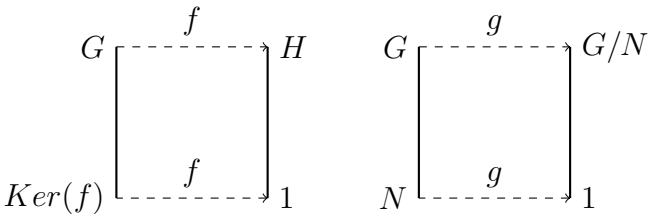

Figure 1.3.1: Group homomorphisms

1.3.2 Rings and fields

Definition 1.3.7. Given a set G and two binary operations $\circ$ and $\star$, the pair $(G, (\circ, \star))$ is denominated ***ring*** if $(G, \circ)$ forms an Abelian group and $\star$ is such that the following properties are valid.

1. **Closure.** If $a \in G$ and $b \in G$, then $a \star b \in G$.
2. **Distributive law.** For any $a, b, c \in G$, then $a \star (b \circ c) = (a \star b) \circ (a \star c)$.

A ring where $a \star (b \star c) = (a \star b) \star c$, for any $a, b, c \in G$ is called ***associative ring***. A ring where there is an element $e \in G$, such that $a \star e = e \star a = a$, for any $a \in G$, is denominated ***ring with identity element***. A ring where $a \star b = b \star a$, for any $a, b \in G$, is called ***commutative ring***.

A ***field*** is a mathematical structure where the four operations are permitted, namely, $+$, $-$, $\times$ and $\div$. Let G^* be the set formed by the elements of G excluding the identity element of operation $\circ$. If $(G^*, \star)$ is an Abelian group and the distributive law, described above, is valid, then $(G, (\circ, \star))$ is a field. Given a subset $H \subset G$, if $(H, (\circ, \star))$ is a field, then H is a ***subfield*** of G. Conversely, G is called ***extension field*** of H.

Example 1.3.4. The set of rational numbers $\mathbb{Q}$, together with usual addition and multiplication, forms a commutative ring, such that, for an arbitrary element $a \in \mathbb{Q}$, its additive inverse is $-a \in \mathbb{Q}$ and its multiplicative inverse is $1/a \in \mathbb{Q}$. Hence $\mathbb{Q}$ is a field. Given a ring R, we can construct an example of non-commutative ring by the utilization of square matrices of size $n \times n$, composed by elements $a_{ij} \in R$, with usual matrix addition and multiplication. The set $\mathbb{Z}_n$, with usual modular addition and modular multiplication, forms a ring, whose additive identity is 0 and multiplicative identity is 1.

Let R be a ring. A subset S of R is a ***subring*** of R if S itself is a ring with respect to the same operations defined over R.

Example 1.3.5. Let $\mathbb{Z}$ be the ring of integer numbers. Then, $2\mathbb{Z} = \{\dots, -4, -2, 0, 2, 4, \dots\}$ is a subring of $\mathbb{Z}$, because according to example 1.3.2, $2\mathbb{Z}$ is an Abelian group with respect to addition. Furthermore, given two elements $a, b \in 2\mathbb{Z}$, we have that $a.b$ is even and hence belongs to $2\mathbb{Z}$. Finally, the distributive law is true because $\mathbb{Z}$ itself is a ring.

Given a ring R, a ***zero divisor*** is a non-zero element $a \in R$, such that there is $b \in R$, $b \neq 0$, such that $a.b = 0$. Let R be a commutative ring. If R contains no zero divisor the ring is called ***integral domain***. The ***characteristic*** of a ring is defined by the smallest integer n, such that

$$\sum_{1}^{n} e = 0,$$

where e is the identity element with respect to multiplication. If there is no such value n with this property, then we say that the ring has characteristics 0.

Let $\mathbb{Z}_p$ be the field from example 1.3.4. Thus, the smallest n such that $n.1 = 0 \pmod p$, is the p itself. Therefore, $\mathbb{Z}_p$ has characteristics p. Let $\mathbb{Q}$ be the field defined in example 1.3.4. We know that there is no such value n such that $n.1 = 0$. Hence, the characteristics of $\mathbb{Q}$ is 0.

Let R be a ring with identity. R may have an element a that has no multiplicative inverse, that is, there is no a^{-1}, such that $a.a^{-1} = 1$. Otherwise, if a has an identity, it is called a ***unit*** in R. For example, in the set $\mathbb{Z}$ of integer numbers, the unities are 1 and -1. In $\mathbb{Z}_p$, for p prime, every element $a \in \mathbb{Z}_p$ has inverse a^{-1} such that $a.a^{-1} \equiv 1 \pmod p$ and therefore every element is a unit.

Ideals

Definition 1.3.8. Given a ring R, a subset I of R is called ***right ideal*** if I corresponds to a subring of R, and for any $x \in I$ and $r \in R$, then $xr \in I$. Given the ring R, the set I of R is denominated ***left ideal*** if I corresponds to a subring of R, and for any $x \in I$ and $r \in R$, then $rx \in I$.

If R is a commutative ring, then every right ideal is equal to the corresponding left ideal and in this case we call it an ***ideal***. A ***proper ideal*** is an ideal that is distinct from the subjacent ring. An ideal I is denominated ***prime ideal*** if for any $a, b \in R$ and $a.b \in I$, then we have that $a \in I$ or $b \in I$.

It is easy to show that $p\mathbb{Z}$ is a prime ideal, if and only if p is prime, because given $n = a.b$, n belonging to $n\mathbb{Z}$, but $a \notin n\mathbb{Z}$ and $b \notin n\mathbb{Z}$. On the other hand, given a prime p and integers a and b such that $a.b \in p\mathbb{Z}$, then $a.b = k.p$, for some integer k. Therefore, $p \mid a$ or $p \mid b$.

Let R be a commutative ring. An ideal I of R is denominated ***principal ideal*** if there is $a \in R$, such that the ideal is generated by multiplying each element from R by a. We say that the ideal is ***generated by*** a and denote it by $I = (a)$.

A function $N : R \to \mathbb{R}^+$, such that $N(0) = 0$, is called ***norm*** over an integral domain R, if the following condition holds: (i) $N(a) > 0$ for all $a \neq 0$; (ii) $N(k.a) = |k|.N(a)$, for any integer k; and (iii) $N(a + b) \leq N(a) + N(b)$ (triangle inequality).

An ***Euclidean domain*** is an integrity domain R such that we can define a division-with-remainder algorithm, namely, given $a, b \in R$, with $b \neq 0$, we can write $a = b.q + r$, where $N(r) < N(b)$. The element r is denominated ***remainder*** and the element q is called ***quotient***. An interesting property, that is easy to demonstrate, is that every ideal I in an Euclidean domain R is a principal ideal. To show that, it is enough to consider the element d of minimum norm in I and show that the following statements are valid: (i) $(d) \subseteq I$ and (ii) $I \subseteq (d)$. The statements together allow us to conclude that $I = (d)$. The statement (i) is simple, because $d \in I$ and I is closed with respect to multiplication. To prove statement (ii), consider any element $a \in I$. Using the division-with-remainder algorithm, we have that $a = d.q + r$. However, by minimality of d, we conclude that $r = 0$. Therefore, $a \in (d)$.

Consider from now on the commutative ring R. An ideal I of R is denominated ***maximal ideal*** if there is no ideal J of R, such that I is a proper subset of J. Moreover, R is called ***principal ideal domain*** if every ideal I of R is a principal ideal. The integers $\mathbb{Z}$ are an example of principal ideal domain, because they form an Euclidean domain.

Definition 1.3.9. Given a ring R and a subset $S = \{x_1, \ldots, x_k \mid x_i \in R\}$, we define the ideal I, ***generated by*** S, as being

$$\{r_1x_1 + \cdots + r_kx_k \mid r_i \in R\}.$$

The subset S can be seen as a basis to the ideal I if $|S|$ is minimal, in other words, if there is no smaller subset that generates the same ideal.

Ring homomorphisms

It is possible to extend the definition of group homomorphisms to ring homomorphisms:

Definition 1.3.10. Given two rings R and S, where $+_R$, $+_S$, $\times_R$ and $\times_S$ are the addition and multiplication in R and S, respectively. We say that $f : R \to S$ is a ***ring homomorphism***, if and only if $f(a +_R b) = f(a) +_S f(b)$ and $f(a \times_R b) = f(a) \times_S f(b)$ holds.

The kernel of the homomorphism is also analogously defined as $\mathrm{Ker}(\psi) = \{a \in R \mid \psi(a) = 0\}$. Namely, it is the set formed by the elements $a \in R$ that are mapped to the additive identity in S.

Theorem 1.3.8. Let $\psi : R \to S$ be a homomorphism from R to S. Then $\mathrm{Ker}(\psi)$ is an ideal of R and S is isomorphic to the quotient ring $R/\mathrm{Ker}(\psi)$. On the other hand, if J is an ideal of R, then the map $\psi : R \to R/J$, defined by $\psi(a) = a + J$, for $a \in R$, is a homomorphism whose kernel is J.

Proof. It is analogous to Theorem 1.3.7

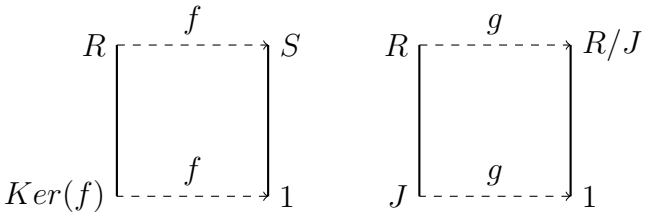

Figure 1.3.2: Ring homomorphisms

Definition 1. *We say that an integer p is **prime** if for integers $a, b \in \mathbb{Z}$ and $p = ab$, then $p|a$ or $p|b$.*

Theorem 1.3.9. Let R be a commutative ring with identity. Then

(i) an ideal I of R is maximal if and only if R/I is a field;

(ii) an ideal I of R is a prime ideal if and only if R/I is an integral domain;

(iii) every maximal ideal is a prime ideal;

(iv) if R is a principal ideal domain, then $R/(c)$ is a field if and only if c is a prime element of R.

Proof. Part (i), (ii) and (iii) are proved respectively in Propositions 12 and 13 and Corollary 14 (Chapter 7) of Dummit and Foote's book [36]. If c is prime, then (c) is maximal and using (i) we conclude that $R/(c)$ is a field. Conversely, we have that if $R/(c)$ is a field, then (c) is maximal and using (iii) we have that c is prime. □

Definition 1.3.11. Ideals R_1 and R_2 in a ring R are ***comaximal*** if $R_1 + R_2 = R$.

The Chinese Remainder Theorem

In this section we describe an important and old theorem, known as the ***Chinese Remainder Theorem*** (CRT). It has important applications in lattice-based cryptography and specially in homomorphic encryption, because it allows to encode information into ***slots*** that can be processed in parallel. The theorem was originally proposed around the third century by Sun Tsu and his famous example asks to find the smallest positive integer number x such that $x \equiv 2 \pmod 3$, $x \equiv 3 \pmod 5$ and $x \equiv 2 \pmod 7$. This system of modular equations can be generalized as follows:

$$\begin{aligned} x &\equiv a_1 \pmod{m_1}, \\ x &\equiv a_2 \pmod{m_2}, \\ &\vdots \\ x &\equiv a_k \pmod{m_k}. \end{aligned}$$

We have that, for every $1 \leq i \leq k$ and any $a_i \in Z_{m_i}$ this equations have an unique solution x modulo $m = \prod m_i$ if and only if $\mathrm{GCD}\,(m_i, m_j) = 1$, for all distinct $1 \leq i, j \leq k$. It is possible to calculate this solution using a constructive method, in resemblance with solving a linear system of equations, where we repeatedly isolate and substitute variables to find the solution. However, it is conceptually better here to study this theorem by considering the homomorphism $f_{\mathrm{CRT}} : \mathbb{Z}_m \to \mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_k}$, given by

$$f_{\mathrm{CRT}}(x) = (x \pmod{m_1}, \ldots, x \pmod{m_k}).$$

Theorem 1.3.10. The map f_{CRT} is a ring homomorphism. Moreover, if $\mathrm{GCD}(m_i, m_j) = 1$ for every (i, j), then f_{CRT} is a bijection, thus it is an isomorphism.

Proof. It is straightforward to show that f_{CRT} is an injective map, since if $f_{\mathrm{CRT}}(x) = f_{\mathrm{CRT}}(x)'$, then we have that $m_i \mid x - x'$ for all $0 \leq i \leq k$. Thus, since all m_i are relatively prime, we conclude that $m \mid x - x'$, therefore $x \equiv x' \pmod{m}$. In order to prove that f_{CRT} is surjective, it suffices to show that the range of f_{CRT} has the same cardinality as the domain, what is trivial, since the cardinality of $\mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_k}$ is equal to $m = \prod m_i$. □

The CRT theorem is not only valid for integers, but it appears also in many other algebraic structures, as for example polynomial rings and number fields. Thus, it is interesting to consider its abstract version, as described in next theorem.

Theorem 1.3.11. Let $R_1, R_2, \ldots, R_k$ be ideals in a ring R. The map $R \to R/R_1 \times R/R_2 \times \cdots \times R/R_k$, defined by $r \to (r + R_1, r + R_2, \ldots, r + R_k)$ is a ring homomorphism with kernel $R_1 \cap R_2 \cap \cdots \cap R_k$. If for any pair of distinct i, j, we have that R_i and R_j are comaximal, then the map is surjective and $R_1 \cap R_2 \cap \cdots \cap R_k = R_1 R_2 \ldots R_k$. Therefore

$$R/(R_1 R_2 \ldots R_k) \equiv R/R_1 \times R/R_2 \times \cdots \times R/R_k.$$

Cyclotomic rings

Definition 1.3.12. We define the ***cyclotomic polynomial*** $\phi_n(x)$ to be the polynomial whose roots are the primitive n-th roots of unity.

Let $n \in \mathbb{Z}$ (usually a power of 2) and consider the ***cyclotomic polynomial*** $\phi_n(x)$, of degree equal to $\varphi(n)$. In the case n is a power of 2, we have that the degree of ϕ_n is equal to $\varphi(n) = n/2$. Given a certain $p \in \mathbb{Z}$, we have that if $\zeta_n \in \mathbb{Z}_p$, where ζ_n is a primitive m-th root of unity in $\mathbb{Z}_p$, then $\phi_n(x)$ can be factored into $\ell = \phi(n)/d$

degree-one polynomials by the Chinese remainder theorem (CRT), for $p^d \equiv 1 \pmod{n}$. Specifically, in the case $d = 1$, we say that the polynomial ***splits completely*** and thus we have that $\phi_n(x) = \prod_{i \in \mathbb{Z}_n^\star}(x - \zeta_n^i)$, where $\mathbb{Z}_n^\star$ is the set formed by elements in $\mathbb{Z}_n$ that are relatively prime to n.

Example 1.3.6. Consider the ring $R = \mathbb{Z}_5[x]/(x^2+1)$. We have that (x^2+1) is the 4-th cyclotomic polynomial and 2 is a primitive 4-th root of unity in $\mathbb{Z}_5$. Thus, we have that $(x^2+1) \equiv (x+2)(x+2^3) \equiv (x+2)(x+3)$.

Let $R = \mathbb{Z}[x]/\phi_n(x)$. Then we call R the ***n-th cyclotomic polynomial ring***. Let $R_p = R/pR$. A polynomial $a(x) \in R_p$ can be represented by a vector of its coefficients in $\mathbb{Z}_p$, called ***coefficients representation***. If indeed $\mathbb{Z}_p$ contains a primitive n-th root of unity ζ_n and if $p^d \equiv 1 \pmod{n}$, we can represent $a(x)$ using evaluations over the distinct primitive n-th roots of unity, given by the powers ζ_n^i, for i an integer prime with n. This representation is called the ***evaluation representation***. Although n must not be restricted to a power of 2, the case $n = 2^k$ is easier to work and have interesting properties for cryptographic usage. Namely, the evaluation representation can be computed using the ***fast Fourier transform*** (FFT) in time $n \log n$. Afterwards, such a representation allows us to compute ring additions and multiplications component-wisely, what means that these operations can be calculated in linear time. Moreover, it can be computed in parallel.

Given an element $a \in R$, it determines an ideal in R and the corresponding ***ideal lattice*** $\mathcal{L}_a$. Such a lattice can be used as the underlying algebraic structure for crytographic constructions, in the sense that breaking the security of the encryption scheme can be shown to be as hard as a certain lattice problem, which is conjectured to be ***computationally hard***, as we are going to see later.

Canonical embedding

It is a common approach in ideal lattice cryptography to represent ideal elements using the ***canonical embedding***, as we are going to describe in this section. This representation is interesting because it offers some advantages, like for example component-wise additions and multiplications by the CRT theorem, a better analysis to the underlying ring ***expansion factor***, which is the measure of how much is the growth of elements after multiplications. Also, it has interesting automorphisms, given by the permutation of the axes of the embedding. Lyubashevsky, Peikert and Regev argue that the canonical embedding in some sense is the right way to represent elements in ideal lattice cryptography [68].

Definition 1.3.13. A ***number field*** K is a field extension of the rationals. Precisely, it is the adjunction to $\mathbb{Q}$ of an abstract element ζ, such that $f(\zeta) = 0$ for a monic $f(x) \in \mathbb{Q}[x]$. The polynomial f is called the ***minimal polynomial*** of ζ and we say that K has degree m, where m is the degree of f.

A number field K can be interpreted as a m-dimensional vector space over $\mathbb{Q}$ with basis given by the powers ζ^i, for $0 \leq i < m$. This basis is called the ***power basis*** of K. Also, we have that the number field K is isomorphic to $\mathbb{Q}[x]/f(x)$. In particular, we have that if $f(x)$ is a cyclotomic polynomial, then ζ is a primitive m-th root of unity.

Definition 1.3.14. Given a number field K, the ***ring of integers*** $\mathcal{O}_K$ is formed by the algebraic integers of K, i.e. by elements whose minimal polynomial has integer coefficients.

Specifically, we are interested in the cyclotomic ring $R = \mathbb{Z}[x]/\phi_n(x)$, where we have that $m = \varphi(n)$, and the power basis $\{1, \zeta, \ldots, \zeta^{m-1}\}$ is an integral basis, thus it is also a basis to the ring of integers $\mathcal{O}_K$.

Definition 1.3.15. Given a degree-m number field K, we have m ring homomorphisms $\tau_i : K \to \mathbb{C}$, for $0 \leq i < m$, mapping ζ to each of the complex roots of the minimal polynomial of ζ. This family of maps gives rise to the ***canonical embedding*** of the number field K, which is defined by the map $\tau : K \to \mathbb{C}^m$, where $\tau(x) = (\tau_0(x), \tau_1(x), \ldots, \tau_{m-1}(x))$.

For cyclotomic rings R, we have that $\tau_i(\zeta) = \zeta^i$, for $i \in \mathbb{Z}_n^\star$. Hence the roots of unity $\tau_i(\zeta)$, for $0 \leq i < m$, have norm equal to 1. The expansion factor improvement is given by $\Omega(\sqrt{n})$ and a complete description of the subjacent mathematics and algorithms to do computations using the canonical embedding is presented in another work of Lyubashesky, Peikert and Regev [69].

1.4 Probability

The study of propability theory is essencial to cryptography and in this section we are going to provide a few results that are important to understand key concepts in lattice-based cryptography, as is the case of the leftover hash lemma.

We denote by $\Pr[\text{event} \mid \text{conditions}]$ the probability that an ***event*** happens given that some ***conditions*** are satisfied. In particular we are only interested in discrete probabilities, which means that all events are sampled from a discrete set S, called ***sample space***. In general, events will be described by binary strings of fixed length, say k, and the sample space S will be given by all possible binary strings of bit length equal to k. Then, we have that $S = \{0,1\}^k$. Any event $a \in S$ have $\Pr[a] \geq 0$. Also, the probability of all the events together sums up to 1, i.e. $\sum_{a \in S} \Pr[a] = 1$. We say that two events a and b are ***independent*** if $\Pr[a \wedge b] = \Pr[a].\Pr[b]$, where $a \wedge b$ symbol is used to denote that both events a and b happens.

A ***random variable*** X is associated to a ***probability distribution*** $\mathcal{D}$ if the probability that the random varialbe X is equal to any event $x \in S$ is determined by $\mathcal{D}$. For

example, the ***uniform*** distribution is the one that gives equal probabilities to every possible event, i.e. $\Pr[X = x] = 1/|S|$.

Definition 1.4.1. The ***expectation*** E of a random variable X is defined by

$$E[X] = \sum_{x \in S} x.\Pr[X = x].$$

In cryptography, we usually want to show that a certain distribution is very close to the uniform distribution. Hence, in order to do that, we need to define how to measure the distance between two distributions.

Definition 1.4.2. Given two statistical distributions $\mathcal{A}$ and $\mathcal{B}$ over the same sample space S, we define their ***statistical distance*** as follows:

$$\frac{\sum_{x \in S} |\Pr[\mathcal{A} = x] - \Pr[\mathcal{B} = x]|}{2}.$$

1.4.1 Important inequalities

Theorem 1.4.1. Boole's inequality (union bound).

$$\Pr[\cup_{i=1}^{n} X_i] \leq \sum_{i=1}^{n} \Pr[X_i].$$

Proof. From set theory, we have that $\Pr[\mathcal{A} \vee \mathcal{B}] = \Pr[\mathcal{A}] + \Pr[\mathcal{B}] - \Pr[\mathcal{A} \wedge \mathcal{B}]$. In general, the following holds:

$$\Pr[\cup_{i=1}^{n+1} A_i] = \Pr[\cup_{i=1}^{n} A_i] + \Pr[A_{n+1}] - \Pr[\cup_{i=1}^{n} A_i \cap A_{n+1}].$$

Since the probability of the last term is a non-negative real value, and repeating the process, we obtain the desired result. □

Theorem 1.4.2. Markov's inequality.

$$\Pr[X \geq c] \leq \frac{E[X]}{c}.$$

Proof. By definition, we have that $E(X) = \sum_{x \in S} \Pr[X = x]x$. We break the summation into two parts to obtain $E(X) = \sum_{x<c} \Pr[X = x]x + \sum_{x \geq c} \Pr[X = x]x$. Again, using the fact that both summations are non-negative real values, we have that $E(x) \geq \sum_{x<c} \Pr[X = x]0 + \sum_{x \geq c} \Pr[X = x]v$. Therefore, we have that $E(X) \geq \Pr[X \geq v]v$, as we need. □

Definition 1.4.3. Given a random variable X, we define the ***variance*** of X by

$$\mathrm{Var}(X) = E((X - E(X))^2).$$

Theorem 1.4.3. Chebyshev's inequality.

$$\Pr[|(X - E(X))| \geq c] \leq \frac{\mathrm{Var}(X)}{c^2}.$$

Proof. Apply Markov's inequality using the definition of variance to get

$$\Pr[|(X - E(X))| \geq c] = \Pr[|(X - E(X))|^2 \geq c^2] \leq E((X - E(X))^2)/c^2,$$

as we want to prove. □

Theorem 1.4.4. Chernoff's inequality. For $1 \leq i \leq n$, let X_i be mutually independent random variables, such that $0 \leq X_i \leq 1$. Define $X = \sum X_i$. Then, for any $c > 1$, we have that

$$\Pr\left[T \geq cE[X]\right] \leq e^{-zE[X]},$$

for $z = c \log c + 1 - c$.

We have that Chebyshev's bound is an improvement to the result achieved by Markov, because by looking at the variance, we have a quadratic dependence on the constant c. If we add the mutual independency constraint, then we obtain a much better bound, given by Chernoff's inequality, which turns out to offer an exponential dependency on c.

1.4.2 Leftover hash lemma

The leftover hash lemma [54] is an important tool in lattice-based cryptography. Essentially, it is used to show that a random combination of public values may have a very close statistical distance to the uniform distribution and this fact is useful to prove the security of cryptosystems.

Definition 1.4.4. Let $\mathcal{H} : \mathcal{D} \to \mathcal{R}$ be a family of hash functions with domain $\mathcal{D}$ and range $\mathcal{R}$. We say that $\mathcal{H}$ is a ***universal family*** of hash functions if $h \in \mathcal{H}$ is a uniformly chosen hash function, then for any $x \neq y$ and $x, y \in \mathcal{D}$ we have that

$$\Pr[h(x) = h(y)] \leq 1/|\mathcal{R}|.$$

A universal hash function is a function whose probability distribution has the property that collisions occur with at most the same probability as the uniform distribution. An stronger property is defined next.

Definition 1.4.5. Let $\mathcal{H} : \mathcal{D} \to \mathcal{R}$ as above. We say that $\mathcal{H}$ is a ***pairwise independent family*** of hash functions if $h \in \mathcal{H}$ is a uniformly chosen hash function, then for any $x \neq y$ and $x, y \in \mathcal{D}$ and any $r_1, r_2 \in \mathcal{R}$ we have that

$$\Pr[h(x) = r_1 \wedge h(y) = r_2] = 1/|\mathcal{R}|^2.$$

If we want to construct a cryptographic object that is indistinguishable from the uniform distribution, like for example to build functions whose output looks like a pseudorandom function, we must describe a mechanism to measure how far some distribution is from uniformity. Next we formalize the notion of closeness to the uniform distribution.

Definition 1.4.6. We say that a distribution is ϵ***-uniform*** if its statistical distance to the uniform distribution is bound above by ϵ.

Theorem 1.4.5. Leftover hash lemma. Let $\mathcal{H} : \mathcal{D} \to \mathcal{R}$ be a pairwise independent family of hash functions. If $h \in \mathcal{H}$ and $x \in \mathcal{D}$ are uniformly and independently chosen, then $h, h(x)$ are $1/2\sqrt{|\mathcal{R}|/|\mathcal{D}|}$-uniform over $\mathcal{H} \times \mathcal{R}$.

1.5 Lattices

In this section we give the main definitions and concepts about lattices. This area of mathematics is also known as ***geometry of numbers*** and was started by Hermann Minkowski in the end of nineteenth century. We also will present hard problem over lattices which are important for cryptography since they allow ***worst-case*** reductions, as we are going to detail later. Lattices are also important to cryptography because they are part of the ***post-quantum cryptography***, due to the fact that quantum computers can not solve, at least with asymptotic gain over classical computers, some problems over lattices that we are going to describe in this section. Such problems can be reduced to intermediate problems, such as SIS and LWE, which are the base of many cryptosystems.

Definition 1.5.1. Formally, lattices are defined as a linear combination of n elements $b_1, \ldots, b_n \in \mathbb{R}^n$, linearly independent, denominated ***lattice basis***.

$$\mathcal{L}(b_1, \ldots, b_n) = \left\{ \sum_{i=1}^{n} x_i b_i \mid x_i \in \mathbb{Z} \right\}.$$

In other words, a lattice is a discrete vector space, i.e. there is an analogy that allows us to use concepts like norm, dimension, orthogonality, linear transformation,

etc. An alternative approach is the utilization of matrix notation, where the basis is represented by a matrix $B = [b_1, \ldots, b_n]$, that belongs to $\mathbb{R}^{n\times n}$. The lattice generated by matrix B is defined by $\mathcal{L} = \{Bx \mid x \in \mathbb{Z}^n\}$, such that the determinant $\det(B)$ is independent from basis choice and corresponds geometrically to the inverse of lattice points density in $\mathbb{Z}^n$.

Definition 1.5.2. Given a lattice $\mathcal{L}(B)$, the vectors that constitute the lattice basis can be interpreted as edges of a dimension-n parallelepiped. Thus, we can define $\mathcal{P}(B) = \{Bx \mid x \in [0,1)^n\}$, denominated *fundamental domain* of B. We can define another parallelepiped such that we have a symmetric region. In order to do that, let $\mathcal{P}_{1/2}(B) = \{Bx \mid x \in (-1/2, 1/2]^n\}$, denominated *centralized fundamental domain* of B.

Theorem 1.5.1. Let $\mathcal{L}(B)$ be a dimension-n lattice and let $\mathcal{P}(B)$ be its fundamental domain, then given an element $w \in \mathbb{R}^n$, we can write w in the form $w = v+t$, for unique $v \in \mathcal{L}(B)$ and $t \in \mathcal{P}(B)$. This equation can be interpreted as a modular reduction, where the vector t is the result of $w \pmod{\mathcal{P}(B)}$.

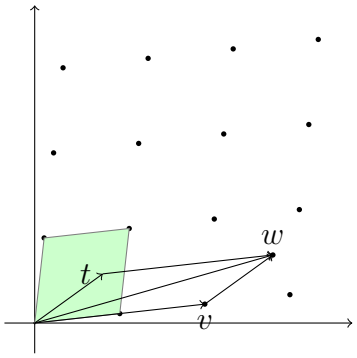

Figure 1.5.3: Reduction modulo $\mathcal{P}(B)$

The volume of the fundamental domain is given by $\mathrm{Vol}(\mathcal{P}(B)) = |\det(B)|$. Given two basis $B = \{b_1, \ldots, b_n\}$ and $B' = \{b'_1, \ldots, b'_n\}$ of the same lattice $\mathcal{L}(B)$, we have that $\det(B) = \pm\det(B')$.

Definition 1.5.3. A ***q-ary*** lattice is defined as the set $\mathcal{L}_q(A) = \{y \mid \exists s \in \mathbb{Z}^n \wedge y = As \pmod q\}$. A ***orthogonal*** lattice is the one obtained by computing a basis composed by vectors that are orthogonal to the original basis. A ***dual*** lattice is defined as the set $\mathcal{L}(A)^* = \{y \mid \langle x, y\rangle \in \mathbb{Z}, \forall x \in \mathcal{L}(A)\}$. It is easy to show that the following relations are valid for dual, orthogonal and q-ary lattices:

$$\begin{aligned} \mathcal{L}_q^{\perp}(A) &= \{y \mid Ay = 0 \pmod q\}, \\ \mathcal{L}_q^{\perp}(A) &= q\mathcal{L}_q(A)^*, \\ \mathcal{L}_q(A) &= q\mathcal{L}_q^{\perp}(A)^*. \end{aligned}$$

In Figure 1.5.4 we illustrate an example of dual lattices. In black we have a lattice given by the basis vectors $(0, 3)$ and $(1, 2)$, while its dual lattice is represented in red and its basis vectors are given by $(1, 0)$ and $(-2/3, 1/3)$.

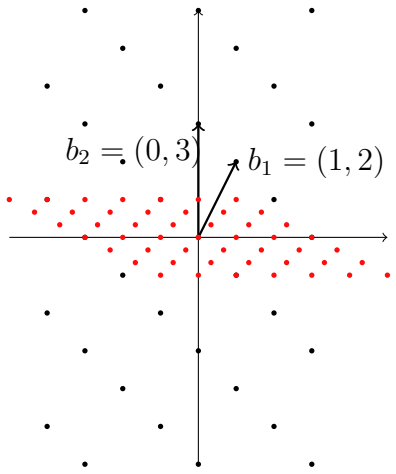

Figure 1.5.4: Dual lattices

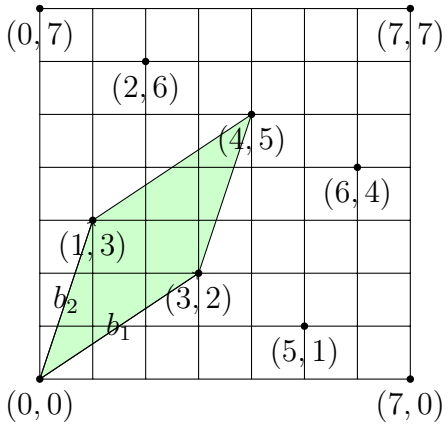

Figure 1.5.5: Q-ary lattice

A q-ary lattice can be represented by a $q \times q$ grid, as shown in figure 1.5.5. Although there is just a limited amount of points inside this region, if we place many copies of it side by side we obtain a regular lattice, with the same basis, but without reduction modulo q.

1.5.1 Hard lattice problems

The problem of finding the shortest vector in a lattice, called the ***shortest vector problem*** (**SVP**) is a fundamental question in lattices. Rigorously, given a lattice $\mathcal{L}(B)$, we wish to find a non-zero vector with minimum norm. This problem can be studied considering two perspectives:

- **search problem:** find a non-zero lattice vector such that its distance from origin is minimized;

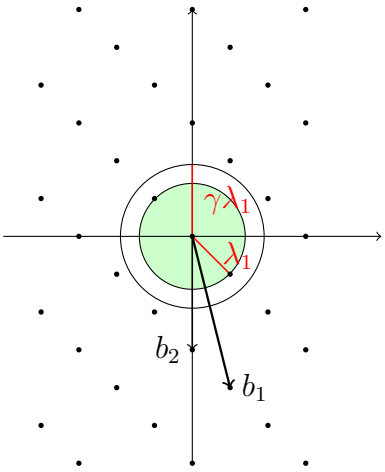

Figure 1.5.6: GAP**SVP**$_\gamma$ example

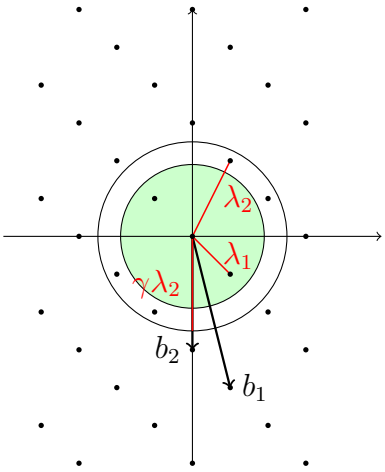

Figure 1.5.7: GAP**SIVP**$_\gamma$ example

- **decision problem:** given a certain norm, determine if there is a vector or not whose length is less than or equal to that norm.

An algorithm to solve the search problem can be used to solve the corresponding decision problem. Moreover, hardness results for the decision problem implies hardness of the search problem. Hence, we can focus on decision problem and if not explicitly mentioned, notation **SVP** refers to the decision version. In practice an approximation factor $\gamma(n)$ is used, in other words we want to decide if there is a vector whose norm is inferior to a certain norm multiplied by $\gamma(n)$. Thus, lattice problems can be studied in the context of ***promise problems***, in which instances are guaranteed to belong or not to a determined subset of all possible instances. In this sense, we denote by GAP**SVP**$_\gamma$ (where n is omitted in order to maintain a cleaner notation), this promise problem using the approximation factor $\gamma(n)$.

Ajtai proved that **SVP** is NP-hard for a random class of lattices [4]. In 1998, Micciancio [71] proved that GAP**SVP**$_\gamma$ is NP-hard for an approximation factor inferior to $\sqrt{2}$, using Euclidean norm. Later, the approximation factor was improved to obtain $\gamma(n) = n^{O(1/\log\log n)}$ [57]. On the other hand, for approximation factors greater than $\sqrt{n/\log n}$, there are strong evidences that GAP**SVP**$_\gamma$ is not NP-hard [3].

Other lattice problems are important for cryptography, as for example:

- the ***closest vector problem*** (**CVP**). Given a lattice $\mathcal{L}(B)$ and a vector $t \in \mathbb{R}^m$, the goal is to find the vector $v \in \mathcal{L}(B)$ closest to t. If we have a bound on the distance from t to the lattice, then the problem is called ***bounded distance decoding*** (**BDD**) problem, as shown in figure 1.5.8;

- and the ***shortest independent vector problem*** (**SIVP**). Given a basis $B \in \mathbb{R}^{n\times n}$, the problem consists in finding n linearly independent vectors $(v_1, \ldots, v_n)$, that belong to the lattice, such that the maximum norm among vectors v_i is minimum, as shown in figure 1.5.7.

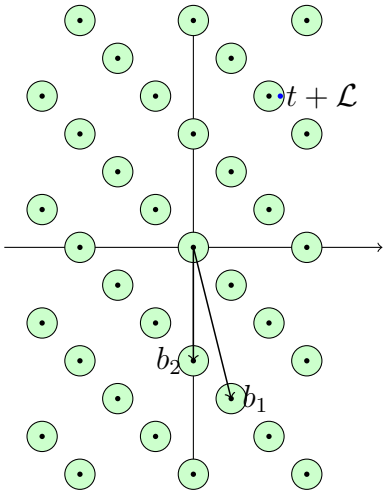

Figure 1.5.8: BDD example

1.5.2 LLL algorithm

In order to solve lattice problems, Babai proposed two algorithms [8], called respectively ***rounding off*** and ***nearest plane***, that essentially proceed in the "obvious" way, but with different strategies. In the first one, it is necessary to solve a system of equations and round each obtained coordinate to the nearest integer, while the second one executes sequentially, reducing the problem in dimension n to a problem in dimension $n-1$, further calling the algorithm recursively to obtain the desired solution. However, these algorithms achieve only approximation factor equal to c^n [73], for a small constant c. Nevertheless, this algorithms are important in lattice-based cryptography, because they can be used to determine concrete parameters to the cryptosystems. In general, as part of the public parameters, we have a basis to the underlying lattice, and this basis is computed using a method that turns it impossible to utilize Babai's algorithms to solve the subjacent lattice problems. The reason is that such a basis is composed by long vectors, that are not sufficiently orthogonal to each other, what makes the rounding errors grow beyond the capacity of Babai's algorithms. Hence, first of all it is necessary to run another algorithm to transform the given basis into a new basis, that makes it possible to have lower rounding errors. Such an algorithm is called ***basis reduction***.

In 1982, in a seminal work, Lenstra, Lenstra and Lovász [61] proposed a basis reduction that became famous as the ***LLL algorithm***. It can be used to solve lattice problems within exponential approximation factors $\gamma(n)$. The exact decisional shortest vector problem is known to be NP-hard, indeed if $\gamma(n)$ is less than or equal to $2^{\log n^{1-\epsilon}}$, then GAP$\mathbf{SVP}_\gamma$ is still NP-hard. For cryptographic purposes, we have that $\gamma(n)$ is given by a polynomial function and therefore the LLL algorithm can not be used to solve underlying lattice problems as indicated in Figure 1.5.9.

Lagrange solved the basis reduction problem for lattices of dimension 2 [82]. Al-

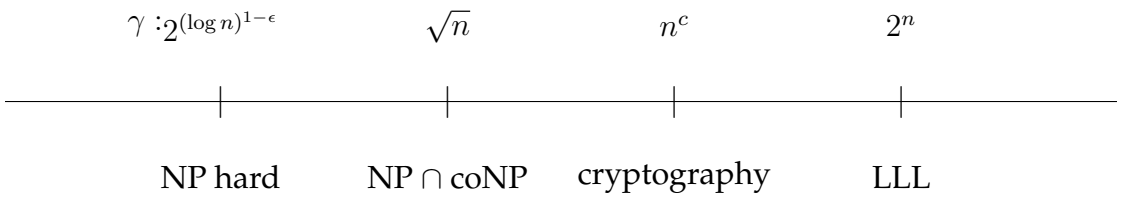

Figure 1.5.9: GAP**SVP**$_\gamma$ complexity

gorithm 1.1 shows how to compute optimal basis for such lattices. The LLL algorithm follows the same ideas of Lagrange's reduction, generalizing and relaxing them in order to obtain a polynomial time algorithm for large dimension n.

Algorithm 1.1 Gauss reduction

INPUT A basis (v_1, v_2).
OUTPUT Returns a basis with shortest vector $(v_1^\star)$ and with a vector $v_2^\star$ that can not be reduced be subtracting v_1.
  $v_1^\star = v_1$ and $v_2^\star = v_2$.
  while true **do**
    if $||v_2^\star|| < ||v_1^\star||$ **then**
      Change $v_1^\star$ with $v_2^\star$.
    Compute $m = \lfloor v_1^\star.v_2^\star / ||v_1^\star||^2 \rceil$.
    if $m = 0$ **then**
      return $(v_1^\star, v_2^\star)$.
    Change $v_2^\star$ with $v_2^\star - mv_1^\star$.

The LLL algorithm in some sense generalizes the Euclidean algorithm to calculate the GCD. The algorithm works based on two main steps: (i) ***size reduction***, where a vector v_i is linearly transformed into another vector that is closer to the hypherplane defined by the basis of the sublattice generated by the vectors $v_1, \ldots, v_{i-1}$; and (ii) ***Lovász condition***, that verifies if v_i is bigger than v_{i-1} multiplying by a constant $\delta = 3/4$. In special, second condition is important in order to obtain a polynomial time algorithm.

Algorithm 1.2 LLL reduction

INPUT Lattice basis $V = [v_1, \ldots, v_n]$.
OUTPUT Returns a lattice basis $V^\star = [v_1^\star, \ldots, v_n^\star]$.
  for $i = 1$ till n **do**
    for $i - 1$ till 1 **do**
      $v_i^\star = v_i^\star - c_{i,j} v_j^\star$ and $v_i^\star$ where $c_{i,j} = \lfloor \langle v_i^\star, v_j^\star \rangle / \langle v_j^\star, v_j^\star \rangle \rceil$
  if $\delta ||v_i^\star||^2 > ||\pi_i(v_{i+1}^\star)||^2$ **then**
    swap $v_i^\star$ and $v_{i+1}^\star$ and repeat.
  else
    return $V^\star$.

There are variations of the LLL algorithm described in the literature. The BKZ-β algorithm uses a subroutine to enumerate short vectors of a sublattice of small dimension β. Then, by combining with the LLL algorithm, although the enumeration considerably increases the running time, it is possible obtain a better basis. To measure the quality of the lattice reduction algorithm $\mathcal{A}$, it is useful to utilize the ***Hermite factor***, denoted by $\delta_{\mathcal{A}}$. This parameter respects the following inequality:

$$||v_1|| = \delta_{\mathcal{A}}^n \det(V)^{1/n}. \tag{1.1}$$

We have that the LLL algorithm achieves $\delta_{\text{LLL}} = 1.021$ and the BKZ algorithm with window size equal to 20 achieves $\delta_{\text{BKZ}-20} = 1.013$. Recently, many proposed improvements were implemented in BKZ algorithm, giving rise to the BKZ-2.0 algorithm [27], which can deal with window size bigger than 50 and whose Hermite factor achieves $\delta_{\text{BKZ}-2.0} = 1.007$. The Hermite factor is essencial in order to estimate the amount of operations that are necessary to break a determined cryptosystem, thus it is a crucial value that must be considered to instantiate a concrete construction of a lattice-based cryptosystem that reaches a certain security level.

1.5.3 Smoothing parameter

An important concept in lattice-based cryptography is the ***smoothing parameter***. It is a very useful lattice invariant, because it allows to erase the discrete structure of a lattice, by making it hard to distinguish between a blurred lattice point and a totally random element [86].

Definition 1.5.4. The Gaussian function $\rho_\sigma : \mathbb{R}^n \to \mathbb{R}^+$ is defined by

$$\rho_\sigma(x) = e^{\frac{-\pi||x||^2}{\sigma^2}}.$$

Given a lattice $\mathcal{L} \subseteq \mathbb{R}^n$, we define the Gaussian distribution $\mathcal{D}_{\sigma,c+\mathcal{L}}(x)$, for any coset $c+\mathcal{L}$, to be zero if x does not belong to the coset $c+\mathcal{L}$ and $\mathcal{D}_{\sigma,c+\mathcal{L}}(x) = \rho_\sigma(x)$, otherwise.

For cryptographic usage, we must be able to sample from Gaussian distributions with high precision. The following list presents the strategies that can be used to accomplish this task:

- ***rejection sampling.*** It was proposed in 2008 [48] and it works by uniformly choosing an element x in the domain of the Gaussian function and then accepting $\rho_\sigma(x)$ with proportional probability, this process is repeated while x is rejected. This strategy requires to compute exponentials by the utilization of float point arithmetic, which is computationally expensive;

- ***inversion method.*** In 2010 [85], Peikert proposed using the inversion method, where we precompute the values $p_z = \Pr[x \leq z : x = \mathcal{D}_\sigma]$ and to sample the Gaussian we can simply generate a uniform element $u \in [0, 1)$ and use binary search to find z such that $u \in [p_{z-1}, p_z)$. The algorithm then outputs z. This strategy requires large tables to store the precomputed data. In 2014, Galbraith and Dwarakanath combined this method with Knuth-Yao algorithm to obtain smaller tables. An adaptation of Ziggurat algorithm [25] is used to achieve a time-memory tradeoff. The idea is to store coordinates of rectangles of the same area, such that it is possible uniformly choose a rectangular and then use rejection sampling to sample a Gaussian inside the retangle.

A problem with this strategies is that both are not appropriate for ***constrained*** devices [37]. While the first one requires expensive float point arithmetic, the second one requires too large tables. A sampling algorithm for a Gaussian over the integers, similar to the Ziggurat approach, and avoiding both mentioned problems appeared in the BLISS digital signature paper [35] and, compared to many alternatives, this proposal came up to be the best choice [83].

Next we give the formal definition of the smoothing parameter and some theorems.

Definition 1.5.5. Formally, given lattice $\mathcal{L}$ and its dual $\mathcal{L}^\star$, the smoothing parameter $\eta_\epsilon(\mathcal{L})$, for $\epsilon > 0$, is the minimal σ such that $\rho_{1/\sigma}(\mathcal{L}^\star) \leq 1 + \epsilon$.

Hence, by perturbing a lattice point using a Gaussian distribution with standard deviation bigger than the smoothing parameter we obtain cosets whose Gaussian mass are equal, except for a small error.

Theorem 1.5.2. [75] For any full rank lattice $\mathcal{L} \subseteq \mathbb{R}^n$, we have that

$$\eta_{2^{-n}}(\mathcal{L}) \leq \sqrt{n}/\lambda_1(\mathcal{L}^\star).$$

There are related theorems in the literature [85,87], depending upon the underlying algebraic structure, the subjacent norm and on the specific property that we want the lattice to respect. For example, as a special case, we have Theorem 1.5.3

Theorem 1.5.3. [75] For any ϵ, we have that

$$\eta_\epsilon(\mathbb{Z}^n) \leq \sqrt{\log\left(2n(1 + 1/\epsilon)\right)/\pi}.$$

1.6 Lattice-based cryptography

In 1996, Ajtai proved NP-hardness of lattice problems [4], showing that a solution to average case instances could be used to find a solution in the worst case. One

year later, the construction of Ajtai-Dwork cryptosystem [5] was proposed, playing an special role in cryptography, because it was the first construction based on ***worst case assumptions***. Other cryptographic primitives were suggested following Ajtai's work, but performance, and in particular the public key size, was not good enough to be used in practice. On the other hand, GGH and NTRU are cryptosystems that have no security proof, but can be efficiently implemented [12].

In this section, we are going to describe cryptographic hash constructions, encryption schemes, digital signatures and other cryptographic primitives that can be built based on assumptions over lattice problems.

1.6.1 Lattice-based hash

The first lattice-based cryptographic primitive to appear in the literature was proposed by Ajtai [4]. It was the appearance of ***worst case reductions***, where an attack to the cryptosystem can be used to solve any instance of hard problems over lattices. In particular, finding collisions for the proposed hash function has in average the same complexity as the SVP problem in the worst case wih respect to the subjacent dual lattice.

Concretely, given $n, m, d, q \in \mathbb{Z}$, we build a cryptographic hash family, $f_A : \{0, ..., d-1\}^m \to \mathbb{Z}_q^n$, indexed by matrix $A \in \mathbb{Z}_q^{n \times m}$. Given a vector $y \in \mathbb{Z}_d^m$, we have that $f_A(y) = Ay \pmod q$. Algorithm 1.3 describes the details invovlved in this operations. A possible parameter choice is given by $d = 2, q = n^2, m \approx 2n \log q / \log d$, such that the hash function has compression factor equal to 2.

Definition 1.6.1. Given the matrix $A \in \mathbb{Z}_q^{n \times m}$, the ***short integer solution*** (SIS) problem is to find short, say binary, vector x such that $Ax \equiv 0 \pmod q$.

It is possible to prove that if one can solve the SIS problem, then we can use this solution to solve any instance of problems like GAP**SVP**$_{\gamma(n)}$ and GAP**SIVP**$_{\gamma(n)}$, for a polynomial approximation factor $\gamma(n)$ [12].

Note that any solution to the SIS problem can be used to generate collisions to the hash family defined above. Indeed, the scheme's security follows from the fact that if one is able to find a collision $f_A(y) = f_A(y')$, then immediately we have that it is possible to compute a short vector in the dual lattice, namely $y - y' \in \mathcal{L}_q^*(A)$.

Algorithm 1.3 Ajtai's hash

INPUT Integers $n, m, q, d \geq 1$. A matrix A chosen uniformly in $\mathbb{Z}_q^{n \times m}$. A vector $y \in \{0, ..., d-1\}^m$.
OUTPUT A vector $f(y) \in \mathbb{Z}_q^n$.
return $f(y) = Ay \pmod q$.

This proposal is really simple and can be efficiently implemented, however in practice, hash functions are designed in an ***ad-hoc*** way, without theoretical garanties provided by a security proof, what allows to obtain faster algorithms than Ajtai's construction. Moreover, if an attacker has access to sufficiently many hash values, then it is possible to recover the fundamental domain of $\mathcal{L}_q^*(A)$, allowing us to compute collisions easily.

In 2002, Micciancio used cyclic lattices to obtain more efficient hash construction. Using this idea we can define the ***ring SIS*** problem analagously to Definition 1.6.1, but with matrix $A \in R^{1\times m}$, for $R = \mathbb{Z}_q[x]/(x^n - 1)$, and such that we are asked to find short x such that $Ax \equiv 0 \pmod q$.

In 2011, Stehlé and Steinfeld [95] proposed a collision-resistent hash function family with better performance, whose construction will be important to digital signature schemes, as we are going show in Section 1.6.3.

1.6.2 Lattice-based encryption

In last section we have seen that it is possible to construct collision resistant hash functions on the assumption that the SIS problem is hard. Moreover, since we have a worst-case reduction from lattice problems to the SIS problem, then we can consider the SIS problem as an intermediate problem under which we are going to base our cryptographic constructions. In this section, we will introduce another important problem. It is called the ***learning with errors*** (LWE) problem and it also has worst case connection to lattice problems. Hence, we are going to see that both SIS and LWE can be used in the design of lattice-based cryptosystems.

GGH

GGH cryptosystem [51] allows us to easily understand the utilization of lattices in public key cryptography. The orthonormality of the basis is a key concept in the design of this cryptosystem, because the private key is defined as a basis B_{priv}, formed by vectors with Hadamard ratio close to 1, meaning that the vectors have good orthonormality. On the other hand, the public key B_{pub} is composed by vectors with Hadamard ratio close to 0, what means that it has not a good orthonormality.

Shorty, the cryptosystem works as follows:

- the encryption algorithm adds noise $r \in \mathbb{R}^n$ to the plaintext $m \in \mathcal{L}$, obtaining the ciphertext $c = m + r$;
- the decryption algorithm must be able to remove the inserted noise. Alternatively, it is necessary to solve an instance of CVP problem.

Figure 1.6.10 shows a dimension 2 lattice, with basis given by vectors v_1 and v_2, almost orthogonal. Figure 1.6.11 shows a different basis to the same lattice, composed by vectors whose Hadamard ratio is close to zero.

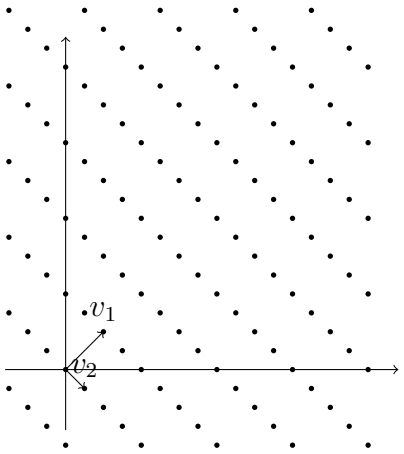

Figure 1.6.10: Good basis

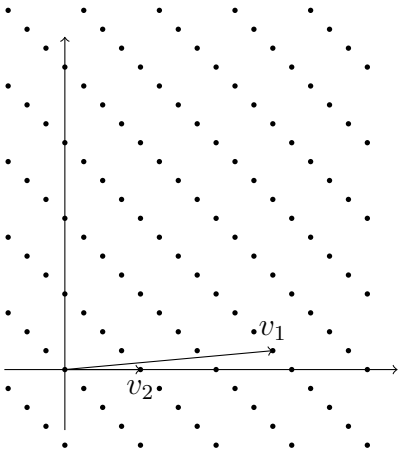

Figure 1.6.11: Bad basis

In high dimension, if basis orthonormality is close to zero, then the CVP problem becomes hard to solve using basis reduction algorithms. Thus we can define the public key as a basis B_{pub}, such that $\mathcal{H}(B_{\text{pub}})$ is close to zero. Furthermore, if we know the private key B_{priv}, then it is possible to use Babai's ***rounding-off*** algorithm [8], defined below in Algorithm 1.4, to recover the plaintext.

Algorithm 1.4 Babai's algorithm

INPUT Dimension n lattice $\mathcal{L}$; a vector $c_{B_{\text{pub}}} = (c_1, ..., c_n)$, where $c_i \in \mathbb{R}$; and a basis $B_{\text{priv}} = (s_1, ..., s_n)$, sufficiently orthonormal.

OUTPUT The vector $m \in \mathcal{L}$ that solves CVP problem with respect to c and $\mathcal{L}$.

Solve the linear system $c_{B_{\text{pub}}} = t_1 s_1 + ... + t_n s_n$, on variables t_i, for $1 \leq i \leq n$.

for $i = 0$ till $i = n$ **do**

$a_i \leftarrow \lfloor t_i \rceil$.

return $m \leftarrow a_1 s_1 + ... a_n s_n$.

The idea of Babai's algorithm is to represent the vector c using the private basis B_{priv}, solving the linear system in n equations. As $c \in \mathbb{R}^{n \times n}$, to obtain a lattice point $\mathcal{L} \subset \mathbb{Z}^n$, each coefficient $t_i \in \mathbb{R}^n$ must be approximated to the nearest integer a_i, where this operation is denoted by $a_i \leftarrow \lfloor t_i \rceil$. This procedure is simple and works very well since basis B_{priv} is sufficiently orthonormal, reducing rounding errors.

One way to attack the cryptosystem is trying to reduce the basis B_{pub}, in order to obtain shorter vector, with Hadamard ratio close to 1. In dimension 2 the problem can be easily solved using Lagrange reduction (algorithm 1.1). For higher dimensions the problem is computationally hard. Unfortunately, this scheme has no security proof and therefore we have no garantees that it is as secure as solving lattice problems.

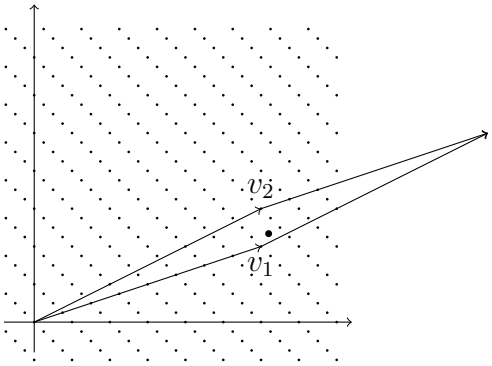

Figure 1.6.12: Bad basis CVP

The NTRU cryptosystem

NTRU cryptosystem [55] is constructed over polynomial rings, but similarly to the GGH scheme we can interpret it as intances of hard problems over lattices, since key recovery and decoding attacks are indeed instances of the SVP and CVP problems, respectively. Hence, the solution to this problems would mean an attack to the cryptosystem, thus we must design our parameters in order to protect against basis reduction algorithms.

The cryptosystem utilizes the following polynomial rings: $R = \mathbb{Z}[x]/(x^n - 1)$, $R_p = \mathbb{Z}_p[x]/(x^n - 1)$ and $R_q = \mathbb{Z}_q[x]/(x^n - 1)$, where n, p, q are positive integers.

Definition 1.6.2. Given positive integers d_1 and d_2, we define $\mathcal{T}(d_1, d_2)$ as the class of polynomials that have d_1 coefficients equal to 1, d_2 coefficients equal to -1 and the remaining coefficients equal to zero. This polynomials are called ***ternary polynomials***.

Definition 1.6.3. The scheme is parameterized by the security parameter λ and the integers n, p, q, d, where n and p are prime numbers, $(p, q) = (n, q) = 1$ and $q > (6d + 1)p$.

Key generation. Choose $f \in \mathcal{T}(d + 1, d)$ such that f has inverse in R_q and R_p. Choose also $g \in \mathcal{T}(d, d)$. Compute F_q as f inverse in R_q and, analogously, F_p the inverse of f in R_p. The public key is given by $h = F_q.g$.

Encryption. Given the plaintext $m \in R_p$ and the public key h, choose randomly $r \in \mathcal{T}(d, d)$ and output $c \equiv pr.h + m \pmod q$.

Decryption. Given the ciphertext c and the secret key f, compute $a = \lfloor f.c \rceil_q \equiv \lfloor pg.r + f.m \rceil_q$. Finally, the message can be obtained computing $m \equiv F_p.a \pmod p$. Output m.

LWE-based encryption

Like GGH, NTRU has not a security reduction to worst-case lattice hard problems. In this section, we are going to present a cryptosystem based on the LWE problem, that is an efficient proposal with security proof based on worst case GAP**SVP**$_{\gamma(n)}$ [89], for a polynomial $\gamma(n)$, where n is the lattice dimension. This proof is a quantum reduction, i.e. it shows that an adversary that has advantage against the cryptosystem implies the existence of a quantum algorithm to solve hard problems over lattices. In 2009, Peikert showed a classical reduction to construct the security proof [84], but under the price of using an exponencial (in the degree n) moduli q.

Definition 1.6.4. The ***LWE problem***, parameterized by n, N, q, σ consists in finding the vector $s \in \mathbb{Z}_q^n$, given equations $\langle s, a_i \rangle + e_i = b_i \pmod q$, for $1 \leq i \leq N$. The values e_i are small errors that were inserted accordingly to the distribuiton $\mathcal{D}_{n,q,\sigma}$, generally taken as an n-dimensional Gaussian distribution over $\mathbb{Z}_q$ with standard deviation given by σ.

In 2010, Lyubaskevsky, Peikert and Regev utilized polynomial rings in their proposal to construct the ***ring LWE*** scheme [68]. By adding algebraic structure to the LWE problem, choosing variable s, a_i and e_i as elements of a determined ring, it is possible to obtain better algorithms and better overhead. Hence, we will focus on this structured version the problem, which is called ***ideal lattice cryptography***. Let $f(x) = x^n + 1$, where n is a power of 2. Given the integer q and an element $s \in R_q = \mathbb{Z}_q[x]/f(x)$, the *ring-LWE problem* over R_q, with respect to the distribution $\mathcal{D}_{n,q,\sigma}$, is defined correspondingly, namely, it is necessary to find s satisfying equations $a_i.s + e_i = b_i \pmod{R_q}$, for $1 \leq i \leq N$, such that a_i and b_i are elements of R_q. Modular reduction on R_q is the same as reducing by the polynomial modulo $f(x)$ and its coefficients modulo q. Also, we denote by a^T the transpose of matrix a.

Definition 1.6.5. The cryptosystem is parameterized by the security parameters λ and the LWE parameters n, N, q, σ. Algorithms KEYGEN, ENC, DEC are defined as follows.

Key generation. The algorithm KEYGEN(1^λ) randomly chooses the vector of polynomials $A \in R_q^N = [a_1, \ldots, a_N]^T$, where $N = n \log q$ and generates $s \in R_q$ and the vector $e \in R_q^m$ using the distribution $\mathcal{D}_{n,q,\sigma}$. The private key is given by $sk = s$, while the public key is given by $pk = (A, b = A.s + e)$. The output is (sk, pk).

Encryption. Given the public key pk and the message $m \in R_2$. Algorithm ENC$_{pk}(m)$ then chooses $e_1, e_2 \in R_q$, using the distribution $\mathcal{D}_{n,q,\sigma}$, randomly chooses the vector of binary polynomials $r \in R_2^N$ and computes (u, v) in the following way:

$$u = A^T.r + e_1 \pmod q,$$
$$v = b^T.r + e_2 + \lfloor q/2 \rfloor.m \pmod q.$$

Decryption. Given the ciphertext (u, v) and the secret key sk, algorithm DEC computes

$$v - u.s = (r.e - s.e_1 + e_2) + \lfloor q/2 \rfloor.m \pmod q.$$

Since the standard deviation of distribution $\mathcal{D}$ is considerably less than the modulus q, we have that $(r.e - s.e_1 + e_2)$ has coefficients whose maximum length considerably less than $q/4$, and each plaintext bit can be computed using a simple computation under each coefficient of the obtained polynomial. If the coefficient is closer to 0 than $q/2$, then the corresponding bit is 0, otherwise it is 1.

Recently, a variation of the NTRU cryptosystem has been proved secure based on the assumption that the LWE problem is hard, allowing us to construct a semantically secure scheme and efficient for lattice-based encryption [95], whose public and private keys, encryption and decryption algorithms has complexity $\tilde{O}(\lambda)$. This asymptotic complexity is remarkable because RSA, ElGamal and ECC requires for example complexity at least $\tilde{O}(\lambda^2)$. If an ideal lattice is used, the public key size is $\tilde{O}(\lambda)$, instead of quadratic in λ, hence using the ring LWE setting is important in order to make lattice-based cryptography practical, but therefore it is crucial to understand clearly the hardness of problems over this specific class of lattices. Interestingly, the SIS problem also can be stated in terms of polynomial rings, giving rise to more efficient cryptosystems [72]. Till now, no attack proposed in the literature has a noticeable advantage when given an ideal lattice, that has more structure, rather than when it receives a general lattice. Although no algorithm performs better to solve ideal lattice problems when compared to the algorithms that solve the conventional LWE problem, many recent results show that for a certain class of ideals, the ring LWE problem is not as hard as expected for cryptographic utilization [38, 52, 70]. Hence, it is crucial to keep investigating the security of ideal lattices, in order to gain confidence that the

mathematical structure that exists in ring LWE can not be used to make it easier to find solutions to the subjacent problems.

The BGV homomorphic encryption scheme [45] is constructed on the assumption that the LWE problem is hard, and it turns out that the LWE problem has been a subject of interest in the cryptographic community in the last years.

1.6.3 Digital signatures

GGH and NTRU cryptosystems can be transformed in order to construct digital signature schemes [12]. However, such proposals are not contemplated with a security proof and, in fact, there are attacks in the literature allowing us to recover the private key given a sufficiently big set of signatures [80], which permits to recover the lattice geometry by computing its fundamental domain.

In 2007, Gentry, Peikert and Vaikuntanathan [49] created a new kind of trapdoor function f, with an extra property: an efficient algorithm that, using the trapdoor, samples elements from the preimage of f. A composition of Gaussian distributions is used to obtain a point close to a lattice vector. This distribution has standard deviation greater than the basis vector within maximum norm, such that the reduction by fundamental domain has distribution that is computationally indistinguishable from the uniform distribution. Furthermore, this construction do not reveal the lattice underlying geometry, because Gaussian distribution is spherical. Given message m and a hash function H that maps plaintexts that belong to the preimage of f, we compute the point $y = H(m)$. The signature is given by $\delta = f^{-1}(y)$. To verify the signature we compute $f(\delta) = H(m)$. This kind of construction was proposed by Bellare and Rogaway [11], using trapdoor permutations and modeling H as a random oracle. Thus, a digital signature scheme is constructed in the existential unforgeability under adaptative chosen plaintext attack model. We use a Gaussian to generate the noise e, such that $f(e) = y$ and $y = v + e$, for a point v chosen uniformly in the lattice. Thus, the construction has a security proof based on worst-case lattice problems.

The constructions presented so far could be described in terms of two functions: $f_A(x) = Ax \pmod q$ - Ajtai's construction, based on SIS problem - and $g_A(s, e) = A^T s + e$ - Regev's construction, based on LWE problem - such that the first function is surjective and the second is injective. In 2012, Micciancio and Peikert [74] showed a simple, secure and efficient way to invert g_A and sample from preimage of f_A, allowing the construction of an efficient digital signature scheme. In this proposal, the Gaussian composition allowed parallelism (in later work [49], and subsequent proposals [95], it was inherently sequential), leading to a concrete improvement. Optimizations described above can be used in applications that are based on function g_A or sampling from preimage of f_A, hence, it is not only important to digital signature, but also to construct encryption schemes that are secure in the adaptive chosen ciphertext attack model.

Another possibility of building digital signatures based on lattice assumptions is

following the Fiat-Shamir paradigm [66,67]. This kind of construction depends on the utilization of a rejection sampling algorithm, that is used to show that breaking the scheme is as hard as solving the SIS problem. In 2013, Ducas et al [35] proposed a variation based on bimodal Gaussians, called BLISS, which improves previous results, but still fails to be competitive with standard solutions such as RSA and ECDSA. For instance, we have that the public key size is equal to 8 KBytes, while RSA has 0.5 KBytes and ECDSA has 0.02 KBytes. However, it is possible to modify the BLISS scheme to obtain better performance and smaller keys. A security analysis were carried out to obtain parameters for different security levels, based on lattice basis reduction BKZ algorithm achieving Hermite factor $\delta = 1.007$. Indeed, a proof-of-concept was implemented and the results was encouraging. It showed that this modified BLISS is in fact competitive with RSA and ECDSA [62].

1.6.4 Other applications

Lattice-based cryptography is interesting not only because it resists to quantum attacks, but also because it have been a flexible alternative to the cosntruction of cryptosystems. In particular, the ring-LWE problem has became more and more important, as it allows us to construct stronger trapdoor functions, with better parameters for both security and performance [74].

Gentry [44] analyzed how flexible a cryptosystem can be, considering not just fully homomorphic encryption, that allows us to compute over encrypted data, but also with respect to access control. Thus, lattice-based cryptography seems to be, according to Gentry, a feasible alternative to explore the limits of cryptomania. Among other applications, it is possible to emphasize the following:

- **multilinear maps**. This is the generalization of the kind of construction that can be achieved with bilinear pairings, that is a map allowing the ***bilinear*** property in its two arguments. This property can be used in different contexts, as for example on identity based encryption. A secure multilinear map construction would be very useful and although every construction proposed till now was attacked [40], it has been object of intense research, because such a primitive would allow the design of new applications;

- **identity based encryption**. For a time, identity based encryption was only achievable by the utilization of bilinear pairings. Using lattices, many proposals were done [14,49], built upon the dual scheme $\mathcal{E}$, which is composed by the algorithms $\mathrm{DualKeyGen}$, $\mathrm{DualEnc}$, and $\mathrm{DualDec}$, as pointed out in Section 1.6.2. Specifically, $\mathrm{DualKeyGen}$ computes the private key as the error e, chosen using the Gaussian distribution, while the public key is given by $u = f_A(e)$. To encrypt a bit b, the algorithm $\mathrm{DualEnc}$ chooses randomly s, chooses x and e' according to the Gaussian and computes $c_1 = g_A(s, x)$ e $c_2 = u^T s + e' + b.\lfloor q/2 \rfloor$. The ciphertext is $\langle c_1, c_2 \rangle$. Finally, $\mathrm{DualDec}$ computes $b = c_2 - e^T c_1$. Then, given the hash function

H, modeled as a random oracle, mapping identities to public keys of the dual cryptosystem, the identity based encryption scheme was constructed as follows:

  - Setup. Choose the public key $A \in \mathbb{Z}_q^{n \times m}$ and the master key as been the trapdoor s, acoording to the description in Section 1.6.3;
  - Extraction. Given the identity id, we compute $u = H(\mathrm{id})$ and the decryption key $e = f^{-1}(u)$, using the trapdoor preimage sampling algorithm with trapdoor s;
  - Encrypt. Given bit b, return $\langle c_1, c_2 \rangle = \mathrm{DualEnc}(u, b)$;
  - Decrypt. Return $\mathrm{DualDec}(e, \langle c_1, c_2 \rangle)$.

- **functional encryption**. Functional encryption is a new primitive in cryptography, that raises new horizons [63]. In this system, a public function $f(x, y)$ determines what the user that knows the key y can infer from a ciphertext, denoted by c_x, according to parameter x. Within this model, who encrypts a message m can previously choose what kind of information is obtained after decryption. Moreover, a trusted party is responsible for key s_y generation, that can be used to decrypt c_x, returned as output for $f(x, y)$, without necessarily revealing information about m. Within this approuch it is possible to define an identity based encryption scheme as a functional encryption special case, such that $x = (m, \mathrm{id})$ and $f(x, y) = m$ if and only if $y = \mathrm{id}$. A recent result [41] proposes the construction of a functional encryption scheme based on lattices, being able to deal with any polynomial size Boolean circuit;

- **attributed based encryption**. This is again a special case of functional encryption, because we can define $x = (m, \phi)$ and $f(x, y) = m$ if and only if $\phi(y) = 1$. Namely, the decryption works if the decrypter's attribute y satisfies the predicate ϕ, such that the encrypter can determine a access control policy (predicate ϕ) for the cryptosystem. There are proposals to achieve this kind of operations based on LWE problem [93] and the multilinear maps construction mentioned above has been used by Sahai and Waters [50] to propose an attributed based scheme for any Boolean circuit, showing one more time that lattice-based cryptography can be somewhat versatile;

- **obfuscation**. There is a negative result proving that obfuscation is impossible in a certain security model. However, lattices where used to construct ***indistinguishability obfuscation***, using a different security model and obtaining a good solution regarding this new model. The construction is based on the LWE problem [41], but it is not yet efficient enough to be used in practice.

1.7 Homomorphic encryption

In 2009, Gentry [43] constructed the first ***fully homomorphic encryption*** (FHE) scheme, solving a conjecture that remained open since 1978 when it was proposed by Dertouzos et al [90]. The cryptographic construction is important because it allows to compute arbitrary algorithms over encrypted data. It is based on hard problems over ideal lattices and hence is part of the ***post-quantum*** cryptography. In this PhD thesis we describe the construction under the assumption that the ***approximate GCD problem*** is hard, following the same blueprint originally described in Gentry's work. We think the AGCD-based scheme allows an easier understanding of the concepts involved in the construction of FHE cryptosystems. Even though no FHE proposal till now turned out to be a practical scheme, we are going to show how to build ***somewhat homomorphic encryption*** (SHE) based on the ***learning with errors*** (LWE) problem, which can be used to evaluate a restricted class of algorithms over encrypted data. We also discuss lattice attacks to homomorphic encryption schemes and how the choice of parameters is determined based on the best-attack effort estimation. We compare variants of the main construction and show that each one has advantages and disadvantages depending on the application and on the concrete scenario under which the cryptosystem is implemented.

After Edward Snowden revelations in 2013, concernment should be growing about how private the internet is and how it will be in the future. It became clear that cryptographic standards where influenced by governments and that companies cooperated in this process of implementing systems that we suppose are secure and private, but in practice turn out to be severely flawed. Privacy is essencial to grant the well-functioning of a democratic state and hence everyone should be worried with the way the internet is being built. Rogaway [92] recently argued that the cryptographic community should be paying more attention to problems related to the privacy of users, rather than to the security of companies.

In the cloud computing scenario, conventional encryption schemes can be used to provide privacy to sensitive information, but normally at the cost of losing basically all the functionality, because data must be in the clear in order to do something interesting with it, as for example computing a function or evaluating an algorithm that receives this data as input. In some sense, cryptography seems to be orthogonal to functionality. However, we are going to show that ***fully homomorphic encryption*** (FHE) allows to compute ***any*** algorithm over encrypted data and hence it is a candidate to solve this problem, since it reaches both privacy and ***maximal*** functionality. Unfortunatelly, the solution has a problem, because it is too expensive in terms of computational resources. In particular, it would be interesting to have the capacity to design a scheme providing limited functionality (instead of maximal), but such that it is enough for an specific purpose. It would allow us to know the minimal overhead in the resources cost when compared to computing over the plaintext data. Then we could decide if this resource cost is affordable or not.

Thus, the construction of FHE is a theoretical big and important advance to cryptology, because many interesting techniques were used and many beautiful ideas and concepts were introduced. Nevertheless, we must know how it could be directly used to increase privacy in practice. Thus, we first need to investigate how to transform FHE into practical cryptosystems. In order to do that we have to understand the inherent tradeoffs among security, efficiency and functionality of homomorphic encryption schemes, where the last one is the measure of how much homomorphic computation can be done over encrypted data. This comprehension allows us to design more efficient systems, but with less functionality. For instance, we will show how to construct schemes that have an upper bound on the number of homomorphic operations that they can deal with. Such schemes are called ***somewhat homomorphic encryption*** (SHE). Therefore, the motivation to study homomorphic encryption is twofold: firstly, by considering the theoretical contributions and new ideas involved in the construction of FHE, we can comprehend how expensive it is to achieve such kind of functionality in the design of cryptographic primitives; secondly, by understanding the hardness of the underlying problems and the algorithms that solve them, we can decide how to choose parameters in order to obtain feasible solutions to be used in practical scenarios.

If Alice and Bob want to communicate over an insecure channel, they should use a symmetric cryptographic scheme to protect exchanged messages against an eavesdropper. In order to do that, they must use a shared secret k, generated by an algorithm called KeyGen, that receives as input the security parameter λ, such that attacking the scheme requires at least 2^λ operations. We can define the domains $\mathcal{K}$, $\mathcal{M}$ and $\mathcal{C}$, respectively as the key space, from where algorithm KeyGen computes its outputs, the plaintext space $\mathcal{M}$ and the ciphertext space $\mathcal{C}$. Furthermore, we can define the encryption algorithm $\text{Enc} : \mathcal{M} \times \mathcal{K} \rightarrow \mathcal{C}$ and the decryption algorithm $\text{Dec} : \mathcal{C} \times \mathcal{K} \rightarrow \mathcal{M} \cup \{\perp\}$, such that $\text{Dec}_k(\text{Enc}_k(m)) = m \cup \{\perp\}$, where the symbol $\perp$ is used to denote the case when the ciphertext is an invalid input to the decryption algorithm.

In 1976, Diffie and Hellman [34] published the famous article ***New directions in cryptography***, introducing the concept of public key cryptography (asymmetric cryptography). In this model, Alice uses the algorithms KeyGen to generate a key pair $(sk_A, pk_A) \in \mathcal{K}_{\text{priv}} \times \mathcal{K}_{\text{pub}}$. The private key sk_A must be maintained secret while the public key pk_A may be published. Encryption and decryption algorithms are defined respectively by $\text{Enc} : \mathcal{M} \times \mathcal{K}_{\text{pub}} \rightarrow \mathcal{C}$ and $\text{Dec} : \mathcal{C} \times \mathcal{K}_{\text{priv}} \rightarrow \mathcal{M} \cup \{\perp\}$, where $\perp$ is used to represent the invalid ciphertext tag, and such that $\text{Dec}_{sk}(\text{Enc}_{pk}(m)) = m$, for (sk, pk) a valid key pair. Namely, it is the output of algorithm KeyGen. The scheme $\mathcal{E} = \{\text{KeyGen}, \text{Enc}, \text{Dec}\}$ is denominated ***asymmetric encryption scheme***.

In the same article, Diffie and Hellman proposed an algorithm to use Alice and Bob key pairs to establish a secret key for conventional (symmetric) cryptography. Given a group G, such that $|G| = n$ and a group generator g, the algorithm KeyGen randomly chooses $a \in [0, n)$, computes $A \equiv g^a \pmod{n}$ and returns $(sk_A, pk_A) = (a, A)$ to Alice.

Analogously, Bob obtains the key pair (b, B), with $b \in [0, n)$ randomly chosen and $B \equiv g^b \pmod{n}$. Alice uses Bob's public key, B and her own private key a to compute

$$B^a = (g^b)^a = g^{ab} \pmod{n}.$$

Similarly, Bob uses Alice's public key, A, and his own private key b to compute

$$A^b = (g^a)^b = g^{ab} \pmod{n}.$$

In this manner Alice and Bob compute the same value, that shall be used as secret key. Ironically, Diffie and Hellman suggested a new form of cryptography, without saying how to construct it, at the same time that they abstractly solved symmetric cryptography most important problem.

Two years later, in 1978, Rivest, Shamir and Adleman [91] constructed a public key cryptosystem, named RSA, using a similar idea. Shortly, given $n = p.q$, where p and q are big prime numbers. The algorithm KEYGEN returns the pair (sk, pk), such that $sk = (d, p, q)$, $pk = (e, n)$ and $d.e \equiv 1 \pmod{\phi(n)}$. The encryption algorithm computes $c = \text{ENC}_{pk}(m) = m^e \pmod{n}$, while the decryption algorithm computes $\text{DEC}_{sk}(c) = c^d \pmod{n}$. Correctness is guaranteed because $\text{DEC}_{sk}(\text{ENC}_{pk}(m)) = \text{DEC}_{sk}(m^e \pmod{n}) = m^{e.d} \pmod{n} \equiv m \pmod{n}$.

In special, given two ciphertexts $c_1 = \text{ENC}_{pk}(m_1)$ and $c_2 = \text{ENC}_{pk}(m_2)$, we have that $c_1.c_2 = m_1^e.m_2^e = (m_1.m_2)^e \pmod{n}$. In general, given k ciphertexts $c_1, \ldots, c_k$, we have that $\prod c_i = \text{ENC}_{pk}(\prod m_i)$. Thus, RSA preserves the structure of multiplication and a natural question that emerges is whether it is possible to obtain an scheme that preserves both multiplications and additions. Mathematically, such a map is called a ***ring homomorphism*** (as in Definition 1.3.10).

Still in 1978, Rivest, Adleman and Dertouzos [90] defined the concept of ***secret homomorphisms*** as being a mapping between algebraic systems, composed by operations, predicates and constants (preserved by the mapping). In other words, it is a cryptographic scheme $\mathcal{E} = \{\text{KEYGEN}, \text{ENC}, \text{DEC}, \text{EVAL}\}$, where the algorithm EVAL is able to evaluate algebraic circuits that belong to a permitted domain, denoted by $\mathbf{S_C}$, composed by additions and multiplications over ciphertexts. Namely, $\text{EVAL} : \mathcal{K}_{\text{pub}} \times \mathbf{S_C} \times \mathbb{C}^k \to \mathbb{C}$, such that for each circuit $\mathbf{C} \in \mathbf{S_C}$, if $\bar{c} = \langle c_1, \ldots, c_k \rangle$ is a vector of ciphertexts such that $c_i = \text{ENC}_{pk}(m_i)$, then we have that $m = \mathbf{C}(m_1, \ldots, m_k)$ and $m = \text{DEC}_{sk}(\text{EVAL}_{pk}(\mathbf{C}, \bar{c}))$. The set of algorithms $\mathcal{E} = \{\text{KEYGEN}, \text{ENC}, \text{DEC}, \text{EVAL}\}$ is called ***fully homomorphic encryption (FHE)***, if $\mathbf{S_C}$ is equivalent to the set of all Boolean circuits. Formally it is necessary to establish conditions in order to the obtain a practical cryptosystem. For example, the ciphertext must not grow too much with respect to the size of the circuit that we want to evaluate. Furthermore, key generation, encryption, decryption and evaluation algorithms must have polynomial complexity with respect to the security parameter. In the same article, the authors proposed some concrete secret homomorphisms, but they were all proved to be insecure.

If $\mathbf{S_C}$ is not equivalent to all Boolean circuits, but contains all algebraic circuits of

multiplicative depth at most equal to some positive integer ℓ, then we say that the scheme is called ***somewhat homomorphic encryption (SHE)***.

1.8 Security model

A cryptosystem is secure against ***chosen ciphertext attack*** (CCA2) if there is no polynomial time adversary that can win the following game with non-negligible probability.

Setup. The challenger obtains $(\mathrm{sk}, \mathrm{pk}) = \textsc{KeyGen}(\lambda)$ and sends pk to adversary $\mathcal{A}$.

Queries. $\mathcal{A}$ sends ciphertexts to the challenger, before or after the challenge. The challenger returns the corresponding plaintexts.

Challenge. The adversary randomly generates two plaintexts $m_0, m_1 \in \mathcal{M}$ and sends them to the challenger, who chooses randomly a bit $b \in \{0, 1\}$ and computes the ciphertext $c = \textsc{Enc}_{\mathrm{pk}}(m_b)$. The challenger sends c to $\mathcal{A}$.

Answer. $\mathcal{A}$ sends a bit b' to the challenger and wins the game if $b' = b$.

If we allow queries only before the challenge, we say that the cryptosystem is secure against CCA1 adversaries (also known as ***lunchtime*** attacks). Queries can be interpreted as an access to a ***decryption oracle***. If instead we only allow access to an ***encryption oracle***, i.e. the adversary can choose any message to be encrypted under the same key pair, then we say that the cryptosystem is secure against ***chosen plaintext attacks*** (CPA).

In homomorphic encryption, it is impossible to achieve CCA2 security, because the adversary can add an encryption of zero to the encrypted challenge, or multiply it by the encryption of one, and send it to the decryption oracle, which allows him to trivially win the game. Many FHE schemes have as public value an encryption of the private key bits, which can be sent to the decryption oracle before the challenge, making such schemes insecure against CCA1 adversaries. Indeed, a ***key recovery*** attack is stronger than a CCA1 attack and Loftus et al [65] showed that Gentry's construction over ideal lattices is vulnerable to it and presented the only SHE proposal that is known to be CCA1-secure.

To prove the security of a cryptosystem, we suppose the existence of an adversary that win the above game and use it to solve some known hard problem. We say that a cryptosystem has security level λ if the best known algorithm to solve the underlying hard problem takes at least 2^λ basic operations to be computed.

Notation remark. Strictly, since the above game asks to distinguish between two cases, in order to obtain a secure scheme we want to prove that ciphertexts are ***indistinguishable*** under CCA1 or CCA2 attacks. Thus, it is usual to refer to this model in the literature by notation IND-CCA1 or IND-CCA2, respectively. However, to simplify notation we are going to denote it by CCA1 and CCA2.

1.9 Intermediate problems

In this section we give intermediate problems, whose hardness is known to be asymptotically at least the same as a determined lattice hard problem. We begin by preseting a simple problem: the ***approximate GCD problem***. Consider the following distribution:

$$\mathcal{D}_{\gamma,\rho}(p) = \{pq + r \mid q \leftarrow \mathbb{Z} \cap [0, 2^\gamma/p), r \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho)\}.$$

Notation $x \leftarrow S$ is used to say that x is uniformly chosen in the finite set S.

Definition 1.9.1. Given parameters (ρ, η, γ), such that $\rho < \eta \ll \gamma$, and a polynomial number of elements from the distribution $\mathcal{D}_{\gamma,\rho}(p)$, for a randomly chosen odd integer p, the ***AGCD problem***, consists in revealing p.

In order to solve this problem we can use many strategies, but in general we have that the LLL algorihtm is a central and fundamental piece to find solutions to the problem. Among the strategies that one could use, we remark the following:

- the Coppersmith's method, as for example in Howgrave-Graham's proposal [56] or in Cohn-Heninger attack [30];
- the utilization of the Lagarias algorithm for simultaneous Diophantine approximations [58];
- using the orthogonal lattice approach [81];
- and also by the utilization of brute force in the noise [26].

A common requirement to resist against these attacks is that $\gamma = \tilde{\Omega}(\eta^2)$, where $\tilde{\Omega}$ notation is used to ignore logarithmic terms. However, it is important to remark that each attack has its peculiarities and the successness depend also upon the relations among η, ρ and λ.

Definition 1.9.2. The ***LWE problem***, parameterized by n, N, q, σ consists in finding the vector $s \in \mathbb{Z}_q^n$, given equations $\langle s, a_i \rangle + e_i = b_i \pmod q$, for a_i public and uniformly chosen in $\mathbb{Z}_q$, $1 \leq i \leq N$. The values e_i are small errors that were inserted accordingly to the distribuiton $\mathcal{D}_{n,\sigma,q}$, generally taken as an n-dimensional Gaussian distribution with standard deviation given by σ.

The LWE problem is interesting because of the following important properties:

Search-to-decision reduction. Given an oracle to solve the decision version the LWE problem, we can solve the search version in polynomial time. The security game

defined in Section 1.8 is a decision problem to the adversary, therefore supposing the existence of such an attacker, we can solve the search version of LWE.

Worst-case connection. Given an oracle the search version of the LWE problem, we can solve one of the lattice problems in the worst case. Thus, we can generate instances of the LWE problem that are hard in the average, under the assuption that it is not possible to solve worst-case instances of problems like the SVP problem.

Quantum resistance. No quantum algorithm is known to offer an asymptotic advantage over classical algorithms that solve this problem. Hence, it can be used to construct ***post-quantum*** cryptography.

Chapter 2

Publications

> An experiment is a question which science poses to Nature, and a measurement is the recording of Nature's answer.
>
> Max Planck

This chapter contains the works related to homomorphic encryption that were published during the PhD program. We remark that other papers were writen during the PhD program [17,18], in the area of elliptic curve crytography, and thus are not present in this thesis.

We also have work related to post-quantum cryptography [10]. My contribution was to write the section of that chapter related to lattice-based cryptography. It describes basic knowledge about lattice-based cryptography and presents a high-level description of some cryptographic primitives, such as: (i) Ajtai's hash-based construction; (ii) the GGH, NTRU and LWE encryption schemes; and (iii) a few comments about lattice-based digital signatures and other interesting applications. These concepts were described in last chapter and thus not going to explore it further. We remark only that the work derives from a short course writen in Portuguese [9]. Moreover, this material has evolved, with new results from the literature and better explanations of fundamental concepts, the result of which is now a technical report [78].

2.1 Homomorphic encryption

In this section we present a short course in Portuguese that was presented in 2012 at the Brazilian Symposium on Information and Computational Systems security, XII SBSeg [33].

In this short course we describe the DGHV construction of FHE and the BGV scheme, that is the state-of-the-art in homomorphic encryption research. Since the paper is somewhat old, we also provide some updates in Chapter 3 of this thesis,

showing new results and describing the attacks that can be used to break these cryptographic schemes.

This section is important because it gives basic definitions that are essential in the study of homomorphic encryption, and could be helpful to students initiating their research in this field. This material was updated and translated to English, giving rise to a technical report [77]

Chapter 1

Encriptação homomórfica

Autores: Eduardo Morais e Ricardo Dahab
Apresentador: Eduardo Morais
Instituição: Unicamp

Abstract

In 1978, Rivest, Adleman and Dertouzos [RAD78] suggested the construction of privacy homomorphisms as a mechanism to protect computation on secret data. The problem remained open till 2009 [Gen09a], when Craig Gentry proposed the utilization of ideal lattices to construct a fully homomorphic cryptosystem.

The usage of privacy homomorphisms is an interesting solution in a cloud computing scenario, but unfortunately it is not efficient enough to be used in practice. However many concrete improvements have been studied recently, encouraging people to look for better algorithms. In this course Craig Gentry's breakthrough will be carefully studied, presenting the state of art and pointing out the problems that remain to be solved.

Resumo

Em 1978, Rivest, Adleman e Dertouzos [RAD78] sugeriram a construção de ***homomorfismos secretos*** - privacy homomorphisms - *como forma de prover um mecanismo de proteção para computação sobre dados sigilosos. O problema permaneceu em aberto até recentemente, quando em 2009 Craig Gentry [Gen09a] o resolveu sugerindo a utilização de reticulados ideais na construção de um criptossistema completamente homomórfico.*

Infelizmente a proposta de Craig Gentry não é eficiente o suficiente para ser usada na prática, mas inúmeros trabalhos têm contribuído para que a eficiência dos algoritmos se torne cada vez maior. Neste minicurso serão estudados os esquemas recentemente propostos por Craig Gentry, apresentando o estado da arte e analisando os problemas que ainda precisam ser resolvidos.

1.1. Introdução

É grande a quantidade de aplicações em nuvem que estão surgindo recentemente e a segurança deste novo modelo de computação ainda é uma questão em aberto. O NIST [MG09] define computação em nuvem, descrevendo três categorias distintas: (i) **software como um serviço** (SaaS - *Software as a Service*) (ii) **plataforma como um serviço** (PaaS - *Platform as a Service*) e (iii) **infraestrutura como um serviço** (IaaS - *Infrastructure as a Service*). Nestes três modelos, a segurança pode ser obtida por meio do uso de criptografia. Além disso, a utilização de uma base de computação confiada (TCB - *Trusted Computing Base*) para gerenciamento de distribuição de chaves, permite a criação de canais seguros entre um cliente e um provedor de serviço em nuvem. Assim, a informação sigilosa pode ser protegida contra um adversário que intercepta as mensagens pela rede. Porém, esta informação ainda pode ser acessada pelo provedor do serviço, o que é uma ameaça em diversos cenários, como por exemplo no caso de informações médicas confidenciais, dados bancários, ou qualquer informação que fira o direito de privacidade de uma pessoa. Portanto, é clara a necessidade da **criptografia como um serviço** (CaaS), podendo ser utilizada nos três modelos de computação em nuvem para prover requisitos como sigilo, autencidade, integridade e não repúdio.

A existência de algoritmos eficientes para obtenção de **homomorfismo secreto** - *privacy homomorphism*, como apresentado por Rivest, Adleman e Dertouzos em 1978 [RAD78], é considerada como o santo graal da criptografia moderna, porque permite a construção de aplicações que em inglês são chamadas de *killer applications*. Estas aplicações podem ser usadas para prover CaaS, podendo citar por exemplo as seguintes possibilidades:

- **banco de dados encriptados**, a nuvem manipula os dados encriptados e retorna para o usuário decriptar;
- **disco rígido encriptado**, semelhantemente ao caso anterior, a nuvem não consegue obter informação confidencial do disco, mas consegue manipulá-lo;
- **mecanismo de buscas encriptado**, permite buscas na internet sem revelar informação sobre o que está sendo buscado;
- **computação sobre dados encriptados**, permite delegar o processamento de um programa à nuvem, que computa sobre os dados encriptados e portanto não tem acesso à informação sigilosa;
- **ofuscação**, embora seja impossível obter ofuscação em um modelo de segurança rígido, veremos como é possível usar homomorfismo secreto para construir um esquema de ofuscação sob a hipótese de uso de um hardware resistente a ataques laterais.

Recentemente [Gen09a], Craig Gentry propôs um esquema baseado em reticulados ideais, cuja ideia é utilizar um ruído r na encriptação, de modo que a

decriptação só funcione caso este ruído seja menor que um determinado limiar $\mathscr{R}$, como ocorre por exemplo no criptossistema GGH [GGH]. É possível efetuar as operações de soma e multiplicação de textos cifrados, alterando o ruído r proporcionalmente a r e r^2, respectivamente para cada operação. Com isso, é possível avaliar circuitos algébricos de profundidade multiplicativa máxima $\log_2(\mathscr{R})$. Para construir um esquema capaz de avaliar circuitos de profundidade arbritária, Craig Gentry alterou esta ideia para que o algoritmo de decriptação pudesse ser expresso como um circuito de baixa profundidade multiplicativa, de modo que fosse possível reduzir o ruído por meio de uma operação que foi denominada ***reencriptação*** (*recryption*).

Para ter segurança equivalente a 2^λ, a performance do esquema proposto por Craig Gentry, após algumas otimizações, é capaz de computar cada operação de um determinado circuito em tempo quase linear em função de λ^6, enquanto a chave pública tem tamanho λ^7. Ao invés de utilizar reticulados, é possível aplicar as mesmas ideias sobre os números inteiros, conduzindo a um esquema com eficiência semelhante, porém com chave pública de tamanho da ordem de λ^{10}. Em trabalhos recentes, o tamanho da chave pública foi reduzido para λ^7 e λ^5 (respectivamente [CMNT11] e [CNT11]).

Implementações recentes mostram que a eficiência do esquema ainda é um ponto crítico, levando 2.2 horas para geração de chaves e 31 minutos para computar a reencriptação, no caso de reticulados ideais [GH11b] e, no caso de inteiros, levando 43 minutos para geração de chaves e 14 minutos e 33 segundos para computar a reencriptação. É importante salientar que esses dados não correspondem a uma segurança equiparável [CMNT11].

Utilizando criptografia parcialmente homomórfica, é possível realizar computações limitadas sobre os dados encriptados. Embora o esquema de Craig Gentry não seja eficiente para ser usado na prática, ele provê uma forma eficiente de computar **parcialmente** (circuitos com profundidade multiplicativa limitada) com os dados encriptados, permitindo a construção de aplicações de grande interesse [NLV11a]. Em especial, uma recente proposta [BGV11] permite a construção de um esquema capaz de avaliar circuitos algébricos de profundidade multiplicativa L em tempo $O(\lambda L^3)$.

1.1.1. Organização deste documento

O minicurso está organizado da seguinte maneira: no restante desta primeira seção serão apresentos alguns fundamentos matemáticos e definições preliminares, também serão mostradas algumas propostas anteriores a construção de Craig Gentry; na seção 2 serão definidos os conceitos básicos, assim como o criptossistema sobre números inteiros de Craig Gentry; na seção 3 será descrito o esquema sobre reticulados ideais; na seção 4 são apresentadas as otimizações, em especial o esquema BGV; na seção 5 são discutidos esquemas práticos que podem ser implementados a partir de criptossistemas parcialmente homomórficos; na seção 6 serão feitas as considerações finais e alguns exercícios são propostos na seção 7.

1.1.2. Fundamentos matemáticos

Nesta seção serão discutidos brevemente os fundamentos matemáticos necessários para compreender as contruções que serão realizadas nas seções futuras.

1.1.2.1. Circuitos

Um ***circuito booleano*** é um conjunto de portas lógicas conectadas por fios. Formalmente, pode ser modelado por um grafo orientado acíclico. O circuito recebe de entrada um conjunto de variáveis booleanas, que são processadas pelas portas lógicas, gerando um conjunto de variáveis booleanas como saída. A profundidade do circuito é a distância entre as entradas e as saídas do circuito. O tamanho do circuito é a quantidade de arestas do grafo.

O modelo de computação baseado em circuitos booleanos é equivalente ao modelo da máquina de Turing, portanto é um modelo completo, capaz de computar um algoritmo arbitrário. Da mesma forma, ***circuitos algébricos*** também correspondem a um modelo computacional completo. Logo, a obtenção de um mapa que simultaneamente seja um homomorfismo e também possa ser usado para criptografia, significa que é possível somar e multiplicar textos cifrados, e portanto é possível computar circuitos algébricos de maneira homomórfica. Como veremos adiante, algumas condições são necessárias para que esta construção seja segura e eficiente.

1.1.2.2. Álgebra abstrata

As construções que serão descritas adiante, farão uso de conceitos matemáticos da álgebra abstrata. Nesta seção serão apresentados alguns destes conceitos de maneira bastante breve. Existem diversos livros que podem ser usados para obter uma compreensão mais aprofundada do assunto, como por exemplo o de Dummit e Foot [DF04].

Um ***anel*** é uma estrutura matemática composta de um conjunto R e duas operações (geralmente $+$ e $\times$), tal que $(R,+)$ é um grupo abeliano, e as operações $+$ e $\times$ estão relacionadas pela propriedade distributiva. Em geral, temos um elemento neutro multiplicativo, mas nem todo elemento de R possui inverso multiplicativo.

Um ***ideal*** I sobre R é um subconjunto fechado em R, de modo que a multiplicação de elementos $i \in I$, por elementos $r \in R$, permanece no subconjunto I, ou seja, $i \times r \in I$. Um exemplo de ideal sobre o anel de inteiros $\mathbb{Z}$ é o subconjunto $p\mathbb{Z}$, para qualquer inteiro p, já que a multiplicação de qualquer múltiplo de p por um elemento arbitrário $r \in R$ continua sendo um múltiplo de p.

Dados dois anéis R e S, um ***homomorfismo*** h é um mapa entre os anéis, que preserva as operações de soma e multiplicação. Isto é, $h(r_1 + r_2) = h(r_1) + h(r_2)$ e $h(r_1 \times r_2) = h(r_1) \times h(r_2)$. Além disso, os elementos neutros aditivo e multiplicativo

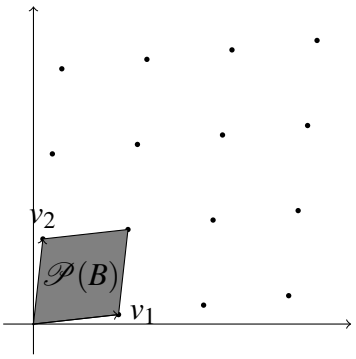

Figura 1.1.

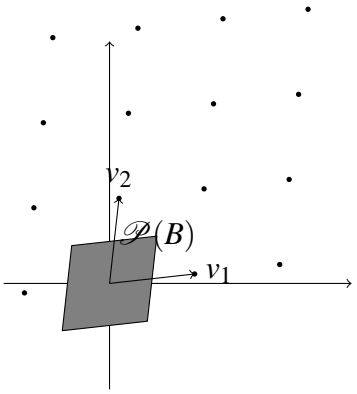

Figura 1.2.

em R são mapeados nos elementos neutros respectivos em S.

1.1.2.3. Reticulados

Reticulados são combinações lineares de n elementos $b_1, ..., b_n \in \mathbb{R}^n$, linearmente independentes, denominados ***base do reticulado***.

$$\mathscr{L}(b_1, ..., b_n) = \left\{ \sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z} \right\}.$$

Em outras palavras, um reticulado é um espaço vetorial discretizado, ou seja, existe uma analogia que nos permite utilizar conceitos como norma, dimensão, ortogonalidade, transformação linear, entre outros. Uma maneira alternativa de abordar o assunto é por meio de notação matricial, onde a base pode ser representada por uma matriz $B = [b_1, ..., b_n]$, pertencente a $\mathbb{R}^{n \times n}$. O reticulado gerado pela matriz B é definido por $\mathscr{L} = \{Bx \mid x \in \mathbb{Z}^n\}$, de forma que o determinante $\det(B)$ é independente da escolha da base e corresponde geometricamente ao inverso da densidade de pontos do reticulado em $\mathbb{Z}^n$.

Dado um reticulado $\mathscr{L}(B)$, os vetores que constituem a base do reticulado são arestas de um paralelepípedo de dimensão n. Assim, podemos definir $\mathscr{P}(B) = \{Bx \mid x \in [0,1)^n\}$, denominado ***paralelepípedo fundamental*** de B. Podemos alternativamente definir $\mathscr{P}_{1/2}(B)$ de forma a obter uma região simétrica. Para isso, seja $\mathscr{P}_{1/2}(B) = \{Bx \mid x \in [-1/2, 1/2)^n\}$, denominado ***paralelepípedo fundamental centralizado*** de B. A figura 1.1 ilustra um exemplo de paralelepípedo fundamental em dimensão 2, enquanto a figura 1.2 representa um paralelepípedo fundamental centralizado.

Seja $\mathscr{L} \in \mathbb{R}^n$ um reticulado de dimensão n e seja $\mathscr{F}$ o paralelepípedo fundamental de $\mathscr{L}$, então dado um elemento $w \in \mathbb{R}^n$, podemos escrever w na forma $w = v + t$, para $v \in \mathscr{L}$ e $t \in \mathscr{F}$ únicos. Esta equação equivale a uma redução modular, onde o vetor t é interpretado como $w \pmod{\mathscr{F}}$.

O volume do paralelepípedo fundamental é dado por $\text{Vol}(\mathscr{F}) = |\det(B)|$. Dadas duas bases $B = \{b_1, ..., b_n\}$ e $B' = \{b'_1, ..., b'_n\}$ de um mesmo reticulado $\mathscr{L}$, temos que $\det(B) = \pm\det(B')$.

O problema de encontrar o vetor de norma mínima (*shortest vector problem* - SVP) é uma das questões fundamentais em reticulados. Rigorosamente, dado o reticulado $\mathscr{L}(B)$, deseja-se encontrar o vetor não nulo com norma mínima. Na prática, é utilizado um fator de aproximação $\gamma(n)$ para o problema SVP, isto é, deseja-se encontrar um vetor cuja norma seja inferior ao vetor de norma mínima, multiplicado por $\gamma(n)$.

Outros problemas em reticulados, importantes do ponto de vista da criptografia, são:

- o ***problema do vetor de distância mínima*** (*closest vector problem* - CVP). Dados um reticulado $\mathscr{L}(B)$ e um vetor $t \in \mathbb{R}^m$, o objetivo é encontrar o vetor $v \in \mathscr{L}(B)$ que seja mais próximo de t;
- e o ***problema dos vetores independentes mínimos*** (*shortest independent vector problem* - SIVP). Dada uma base $B \in \mathbb{Z}^{n\times n}$, o problema consiste em encontrar n vetores linearmente independentes $(v_1, ..., v_n)$, pertencentes ao reticulado, tais que a norma máxima entre os vetores v_i seja mínima.

O criptossistema [GGH97] GGH utiliza o conceito de ortonormalidade da base na definição do par de chaves. A chave privada é definida como uma base B_{priv} do reticulado, formada por vetores quase ortogonais e com norma próxima a 1. Desta forma, antes de prosseguir, é preciso uma forma de medir o grau de ortonormalidade de uma determinada base.

Dado um reticulado $\mathscr{L}$ e uma base $B = (v_1, ...v_n)$, a ***razão de Hadamard***, denotada por $\mathscr{H}(B)$, é definada da seguinte maneira:

$$\mathscr{H}(B) = \left(\frac{\det \mathscr{L}}{||v_1||...||v_n||}\right)^{1/n}.$$

É fácil mostrar que para qualquer base B, temos que $0 \leq \mathscr{H}(B) \leq 1$. Além disso, quanto mais próximo de 1 mais ortonormal é a base. De modo geral, o criptossistema GGH funciona da seguinte maneira:

- o algoritmo de encriptação acrescenta o ruído $r \in \mathbb{R}^n$ ao texto claro $m \in \mathscr{L}$, gerando o texto cifrado $c = m + r$;
- o algoritmo de decriptação precisa ser capaz de retirar o ruído inserido. Alternativamente, é preciso resolver uma instância do problema CVP.

A figura 1.3 mostra um reticulado em dimensão 2, com base dada pelos vetores v_1 e v_2, praticamente ortogonais.

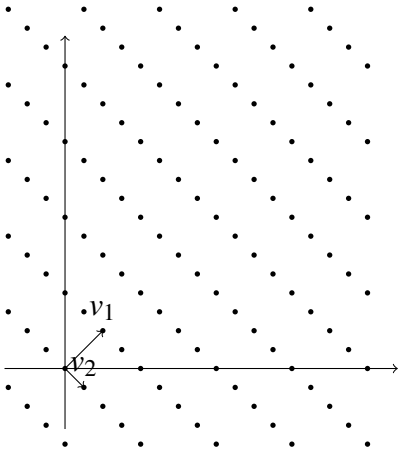

Figura 1.3. Base boa

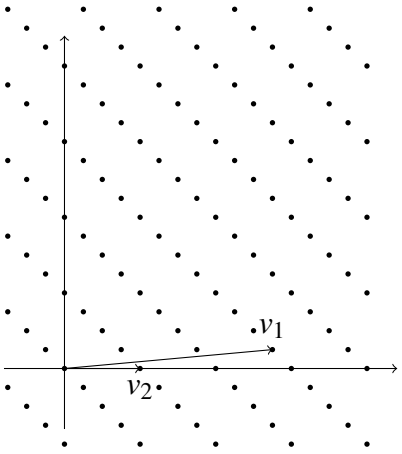

Figura 1.4. Base ruim

Porém, conforme menor a ortonormalidade da base conhecida e maior a dimensão do reticulado, mais difícil é o problema CVP. Desta forma, a chave pública pode ser definida por uma base B_{pub} do reticulado, tal que $\mathscr{H}(B_{\text{pub}})$ seja aproximadamente zero. Por outro lado, com o conhecimento da chave privada B_{priv}, o algoritmo de Babai [Bab86], definido a seguir, pode ser utilizado para recuperar o texto claro.

Algoritmo 1.1.1 Algoritmo de Babai

ENTRADA o reticulado $\mathscr{L}$ de dimensão n; o vetor $c = (c_1, ..., c_n)$, onde $c_i \in \mathbb{R}$; e uma base $B_{\text{priv}} = (s_1, ..., s_n)$, suficientemente ortonormal.

SAÍDA o vetor $m \in \mathscr{L}$ que resolve o problema CVP com relação a c e $\mathscr{L}$.

Resolva um sistema de n equações, $c = t_1 s_1 + ... + t_n s_n$, nas variavéis t_i, onde $1 \leq i \leq n$.

para $i = 0$ até $i = n$ **faça**

$a_i \leftarrow \lfloor t_i \rceil$

retorne $m \leftarrow a_1 s_1 + ... a_n s_n$

A ideia geral do algoritmo de Babai é representar o vetor c na base privada B_{priv}, resolvendo um sistema de n equações lineares. Como $c \in \mathbb{R}^n$, para obter um elemento do reticulado $\mathscr{L}$, cada coeficiente $t_i \in \mathbb{R}$ é aproximado para o inteiro mais próximo a_i, onde esta operação de arredondamento é denotada por $a_i \leftarrow \lfloor t_i \rceil$. Este procedimento simples funciona bem desde que a base B_{priv} seja suficientemente ortonormal, reduzindo os erros do arredondamento.

1.1.3. De 1976 até 2009

No modelo de criptografia convencional (criptografia simétrica), Alice e Bob compartilham uma única chave k, gerada por um algoritmo KeyGen, com a qual podem comunicar-se de forma segura. São definidos os domínios $\mathscr{K}$, $\mathscr{P}$ e $\mathscr{C}$ como sendo respectivamente o espaço de chaves e o espaço de texto claro e texto cifrado. Além disso, são definidos algoritmos de encriptação Enc : $\mathscr{P} \times \mathscr{K} \rightarrow \mathscr{C}$ e

de decriptação $\text{Dec} : \mathscr{C} \times \mathscr{K} \to \mathscr{P}$, tal que $\text{Dec}(k, \text{Enc}(k, m)) = m$. De agora em diante, denominamos o conjunto $\mathscr{E} = \{\text{KeyGen}, \text{Enc}, \text{Dec}\}$ um ***esquema de encriptação simétrica*** ou então um ***criptossistema de chave privada***. Existe um problema imediato neste modelo: a quantidade de chaves que precisam ser gerenciadas é quadrátrica, isto é, em um grupo com n pessoas são necessárias $n(n-1)/2$ chaves para tornar possível a comunicação de quaisquer 2 pessoas deste grupo, de forma que o gerenciamento dessas chaves compartilhadas é um obstáculo a ser superado.

Em 1976, Diffie e Hellman [DH76] publicaram o artigo *New directions in cryptography*, introduzindo o conceito de criptografia de chave pública (criptografia assimétrica). Neste modelo, Alice utiliza o algoritmo KeyGen para gerar um par chaves $(\text{sk}_A, \text{pk}_A) \in \mathscr{K}_{\text{pub}} \times \mathscr{K}_{\text{priv}}$. A chave privada sk_A deve ser mantida em segredo enquanto a chave pública pk_A deve ser divulgada de alguma maneira. Os algoritmos de encriptação e decriptação são definidos respectivamente por $\text{Enc} : \mathscr{P} \times \mathscr{K}_{\text{pub}} \to \mathscr{C}$ e $\text{Dec} : \mathscr{C} \times \mathscr{K}_{\text{priv}} \to \mathscr{P}$, tal que $\text{Dec}(\text{sk}, \text{Enc}(\text{pk}, m)) = m$, para (sk, pk) um par de chaves válido, isto é, gerado por KeyGen. Com estas características, $\mathscr{E} = \{\text{KeyGen}, \text{Enc}, \text{Dec}\}$ é denominado ***esquema de encriptação assimétrica*** ou então ***criptossistema de chave assimétrica***.

Neste mesmo artigo, Diffie e Hellman propuseram um algoritmo que utiliza o par de chaves de Alice e Bob para estabelecer uma chave secreta adequada para criptografia convencional. Dado o grupo G, tal que $|G| = n$ e um gerador g deste grupo. O algoritmo KeyGen gera $a \in [0, n)$ aleatoriamente, calcula $A \equiv g^a \pmod{n}$ e retorna $(\text{sk}_A, \text{pk}_A) = (a, A)$ para Alice. Analogamente, Bob obtém como par de chaves os valores (b, B), com $b \in [0, n)$ escolhido aleatoriamente e $B \equiv g^b \bmod n$. Alice usa a chave pública de Bob, B e sua chave privada a para calcular

$$B^a = (g^b)^a = g^{ab} \pmod{n}.$$

Similarmente, Bob usa a chave pública de Alice, A, e a sua chave privada b para calcular

$$A^b = (g^a)^b = g^{ab} \pmod{n}.$$

Desta maneira Alice e Bob conseguem computar um valor em comum, que pode ser utilizado como chave secreta no modelo de criptografia simétrica. Ironicamente, estavam sugerindo uma nova forma de criptografia, sem dizer quais algoritmos e estruturas matemáticas satisfariam o novo modelo e, além disso, estavam resolvendo o problema de acordo de chaves do modelo antigo.

Dois anos depois, em 1978, Rivest, Shamir e Adleman [RSA83] resolveram o problema desenvolvendo o primeiro criptossistema de chave pública, o RSA, usando uma ideia bem parecida com a que foi apresentada no acordo de chaves de Diffie e Hellman. Resumidamente, $n = p.q$, onde p e q são primos grandes. O algoritmo KeyGen retorna o par (d, e), tal que $d.e \equiv 1 \pmod{\phi(n)}$. O algoritmo de encriptação computa $c = \text{Enc}(m, e) = m^e \pmod{n}$, enquanto o algo-

ritmo de decriptação computa $\text{Dec}(c,d) = c^d \pmod n$. A corretude é garantida porque $\text{Dec}(\text{Enc}(m,e),d) = \text{Dec}(m^e \pmod n, d) = m^{e.d} \pmod n \equiv m \pmod n$.

Em especial, dados dois textos cifrados $c_1 = \text{Enc}(m_1,e)$ e $c_2 = \text{Enc}(m_2,e)$, temos que $c_1.c_2 = m_1^e.m_2^e = (m_1.m_2)^e \pmod n$. Em geral, dados k textos cifrados $c_1,...,c_k$, temos que $\prod c_i = \text{Enc}(\prod m_i, e)$. Assim, o RSA preserva a estrutura da operação de multiplicação e uma pergunta natural que surge é sobre a possibilidade de obter um esquema que preserve ambas as operações de soma e multiplicação. Matematicamente, um mapa assim é denominado ***homomorfismo***.

Ainda em 1978, Rivest, Adleman e Dertouzos [RAD78] definiram o conceito de ***homomorfismos secretos*** - *privacy homomorphisms* - como sendo um mapeamento entre sistemas algébricos, compostos por operações, predicados e constantes (preservados pelo mapeamento). Em outras palavras, é um esquema $\mathscr{E} = \{\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Eval}\}$, onde o algoritmo Eval é capaz de avaliar circuitos algébricos de um domínio permitido, denotado por $\mathbf{S_C}$, compostos pelas operações de soma e multiplicação sobre textos cifrados. Ou seja, $\text{Eval} : \mathscr{K}_{\text{pub}} \times \mathbf{S_C} \times \mathscr{C}^n \to \mathscr{C}$, tal que para cada circuito $\mathbf{C} \in \mathbf{S_C}$, se $\Psi = \langle \psi_1, ..., \psi_n \rangle$ são textos cifrados tais que $\psi_i = \text{Enc}(\text{pk}, m_i)$, então temos que $m = \mathbf{C}(m_1,...,m_n)$ e $m = \text{Dec}(\text{sk}, \text{Eval}(\text{pk}, \mathbf{C}, \Psi))$. O conjunto de algoritmos $\mathscr{E} = \{\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Eval}\}$ é denominado ***criptossistema completamente homomórfico (CCH)***, se $\mathbf{S_C}$ for equivalente ao conjunto de todos os circuitos booleanos. Formalmente é necessário estabelecer condições para que o criptossistema seja prático. Por exemplo, o texto cifrado não pode crescer muito em comparação com o tamanho do circuito que desejamos avaliar. Além disso. os algoritmos de geração de chaves, encriptação, decriptação e avaliação precisam ter complexidade polinomial em relação ao parâmetro de segurança. Estes detalhes serão definidos na seção 1.2.2. Uma nomenclatura alternativa para CCH é ***encriptação completamente homomórfica (ECH)***.

Como vimos, a ideia básica do ***RSA*** [RSA83] pode ser utilizada para construir um criptossistema parcialmente homomórfico, preservando a multiplicação, pois dados os textos cifrados $c_1 = m_1^e \pmod n$ e $c_2 = m_2^e \pmod n$, é possível computar $c_1.c_2 = (m_1.m_2)^e \pmod n$. No próprio artigo de Rivest, Adleman e Dertouzos [RAD78], são propostos esquemas para criação de homomorfismo secreto, mas todos eles foram quebrados.

Uma propriedade importante para a construção de um homomorfismo secreto é a segurança semântica. Se temos conhecimentos de um conjunto $M = \{m_1, m_2, ..., m_k\}$ de textos claros e desejamos saber se um determinado texto cifrado c corresponde a algum m_i e se o algoritmo de encriptação for determinístico, então basta encriptar cada um dos m_i's e comparar o resultado com c. Para ter segurança semântica, um esquema criptográfico deve estar protegido contra este tipo de ataque, e portanto o algoritmo de encriptação deve ser aleatorizado, ou seja, a cada vez que é executado, um novo texto cifrado é gerado, diferente do anterior (com grande probabilidade).

O RSA é um criptossistema determinístico, portanto não possui segurança semântica. Assim, o ***ElGamal*** é uma alternativa imediata, por não ser determinístico e oferecer homomorfismo multiplicativo como o RSA. Dado um número

primo p grande, um gerador g do grupo multiplicativo $\mathbb{Z}_p$, a chave secreta de Alice é um valor a escolhido aleatoriamente entre 0 e $p-1$. A chave pública é dada por $A = g^a \pmod{p}$. Dada uma mensagem $m \in \mathbb{Z}_p$, e um inteiro aleatório k entre 0 e $p-1$, computa-se o texto cifrado como $(c_1, c_2) = (g^k, A^k.m)$. Para decriptar, Alice calcula $m = (c_1^a)^{-1}.c_2 \pmod{p}$. Dados 2 textos cifrados, (c_1, c_2) e (c_1', c_2'), definimos a multiplicação componente a componente, isto é, $(c_1, c_2).(c_1', c_2') = (c_1.c_1', c_2.c_2')$. Sendo assim, é fácil ver que o ElGamal é um homomorfismo, pois $(c_1.c_1', c_2.c_2') = (g^{k_1+k_1'}, g^{a(k_1+k_1')}.m)$. De fato, este homomorfismo mapeia a operação de multiplicação na operação de soma.

O primeiro homomorfismo secreto com demonstração de segurança semântica foi proposto por ***Goldwasser-Micali*** [GM82], usando como base o problema de computar o resíduo quadrático de um elemento de $\mathbb{Z}_N$, onde $N = p.q$, com p e q primos grandes. Calcular o resíduo quadrático em $\mathbb{Z}_p$ ou $\mathbb{Z}_q$ é fácil. Portanto, a chave privada é dada por (p,q), a fatoração de N, enquanto que a chave pública é dada por (N,z), onde z é um elemento de $\mathbb{Z}_N$ tal que $z^{p-1/2} \equiv 1 \pmod{N}$ e z não é um resíduo quadrático em $\mathbb{Z}_N$. Dada uma mensagem $m \in \{0,1\}$, se $m = 0$, o algoritmo de encriptação retorna um resíduo quadrático aleatório em $\mathbb{Z}_N$, caso contrário, se $m = 1$, o algoritmo de encriptação retorna um resíduo não-quadrático c tal que $c^{p-1/2} \equiv 1 \pmod{N}$. A decriptação só pode ser realizada com o conhecimento da fatoração de N, de modo que os resíduos quadráticos possam ser calculados separadamente em $\mathbb{Z}_p$ e $\mathbb{Z}_q$, usando o teorema chinês do resto para calcular o resíduo quadrático em $\mathbb{Z}_N$. Em especial, dados dois resíduos quadráticos, sabemos que a sua multiplicação resulta em um resíduo quadrático. E também é fácil ver que a multiplicação de resíduos não-quadráticos c_1 e c_2, tais que $c_i^{p-1/2} \equiv 1 \pmod{N}$, resulta em um novo elemento $c = \mathbb{Z}_N$, tal que $c^{p-1/2} \equiv 1 \pmod{N}$. Logo, o esquema é homomórfico com relação a multiplicação e pode ser utilizado como homomorfismo secreto.

Particularmente importante é o que criptossistema ***Paillier*** , cuja segurança também é baseada (embora não haja demonstração de que seja equivalente) ao problema de fatoração de um número composto $N = p.q$, com p e q tendo a mesma quantidade de bits. Este esquema utiliza o grupo $\mathbb{Z}_{N^2}^*$. Dado que $N = p.q$, temos que $\mathbb{Z}_{N^2}^*$ é isomorfo a $\mathbb{Z}_N \times \mathbb{Z}_N^*$. De fato, o isomorfismo é dado pela relação $f : \mathbb{Z}_N \times \mathbb{Z}_N^*$, tal que:

$$f(a,b) = (1+N)^a.b^N \pmod{N^2}.$$

A chave pública é o próprio valor de N, enquanto que a chave privada é dada pelo par (p,q). Para criptografar uma mensagem $m \in \mathbb{Z}_N^*$, é computado o valor $c = (1+N)^m.r^N \pmod{N^2}$. Por sua vez, o algoritmo de decriptação computa

$$m = \frac{[c^{\phi(N)} \pmod{N^2}] - 1}{N}.\phi(N)^{-1} \pmod{N}.$$

A encriptação é homomórfica com relação a soma, já que

$$\begin{aligned}\mathrm{Enc}(N,m_1).\mathrm{Enc}(N,m_2) &= ((1+N)^{m_1}r_1^N).((1+N)^{m_2}r_2^N),\\ &= (1+N)^{m_1+m_2 \pmod N}(r_1r_2)^N \pmod{N^2}.\end{aligned}$$

Em geral, a função f é tal que $f(a_1,b_1).f(a_2,b_2) = f(a_1+a_2, b_1.b_2)$.

Outro criptossistema que permite a construção de homomorfismo secreto é o ***Polly Cracker***, proposto por Fellow e Koblitz [FK94], onde um anel polinomial $R = \mathbb{F}_q[x_1,...,x_n]$ contém um ideal I gerado por um conjunto de polinômios públicos, $\{p_1(x_1,...,x_n),...,p_k(x_1,...,x_n)\}$, com uma raíz $\alpha = (\alpha_1,...,\alpha_n)$ em comum, mantida em segredo. Dada uma mensagem $m \in \mathbb{F}_q$, o algoritmo de encriptação computa o polinômio $c(\mathbf{x}) = \sum p_i(\mathbf{x}).r_i(\mathbf{x})$, onde r_i são polinômios escolhidos aleatoriamente, para obteção de um elemento aleatório de I. Para decriptar, basta avaliar o polinômio $c(\mathbf{x})$ em α. A segurança do Polly Cracker é um problema em aberto, porque apesar dos ataques que surgiram, adaptações foram realizadas de modo a sanar as vulnerabilidades.

O criptossistema ***BGN*** [BGN05] é um esquema prático que permite avaliação fórmulas quadráticas, ou seja, permite circuitos com um nível de múltiplicação e um número arbitrário de adições. Sejam $N = p.q$ e considere os grupos $\mathbb{G}$ e $\mathbb{G}_1$, de ordem N, e um emparelhamento bilinear $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$. Dado um gerador $g \in \mathbb{G}$, computa-se $h = g^p$ e a chave pública é dada por (N,h,g), enquanto que a chave privada é a fatoração (p,q) de N. O espaço de texto claro $\mathscr{P}$ é $\mathbb{Z}_p$ e o algoritmo de encriptação computa $c = g^{m+kp}$, com k aleatório e $m \in \mathbb{Z}_p$. O algoritmo de decriptação computa $c^q \equiv g^{mq} \pmod N$ e posteriormente resolve o problema do logaritmo discreto com base g^q. Para que o logaritmo discreto seja eficiente, m precisa corresponder a um elemento de um conjunto com tamanho polinomial, ao invés de ser um elemento qualquer de $\mathbb{Z}_p$. Pelo fato de ser usado o logaritmo discreto como problema difícil subjacente, temos que a multiplicação de textos cifrados corresponde a soma de textos claros. Além disso, dados os textos cifrados $c_1 = g^{m_1+k_1p}$ e $c_2 = g^{m_2+k_2p}$, o emparelhamento bilinear $e(c_1,c_2)$ é igual a $e(g,g)^{m_1.m_2+d.p}$, para um inteiro d.

A segurança de todas as propostas que discutimos anteriormente está relacionada a dificuldade do ***problema de participação em ideal*** (*ideal membership problem*).

Em 2009 [Gen09b], Craig Gentry utilizou reticulados gerados por polinômios ideais para construir o primeiro esquema de ECH, resolvendo assim um problema que ficou em aberto por 31 anos. Devido à complexidade de avaliação das multiplicações e ao tamanho da chave pública, tal proposta ainda não pode ser usada na prática. Porém, otimizações foram propostas, [vDGHV09, SS10, SV09], fazendo-nos acreditar que o ECH está cada vez mais próximo ser se tornar realidade. Com isso, um novo tipo de segurança criptográfica poderá ser oferecido, especialmente no contexto de computação em nuvem. A nova proposta de Craig Gentry está relacionada a um problema ligeiramente diferente, denominado ***problema de classes laterais em ideais*** (*ideal coset problem*).

Em resumo, é possível descrever um modelo genérico do esquema de Craig Gentry como segue:

Geração de chaves. O algoritmo KeyGen escolhe ideais J primo com I e gera as bases B_J^{sk} e B_J^{pk}. Além disso, é determinado uma distribuição $\mathscr{D}_{B_I}(m)$ que gera elementos aleatórios da classe lateral $m+I$.

Encriptação. Dada uma mensagem $m \in R \pmod{B_I}$, utiliza-se a distribuição $\mathscr{D}_{B_I}(m)$ para computar m' e depois é realizada uma redução módulo $\pmod{B_J^{\text{pk}}}$, como segue

$$c = m' = \mathscr{D}_{B_I}(m) \pmod{B_J^{\text{pk}}}.$$

Decriptação. Para decriptar é computado o valor

$$m = [c \pmod{B_{\text{sk}}}] \pmod{B_I}.$$

Graig Gentry utiliza ideais polinomiais para obter um esquema de ***encriptação homomórfica restrita*** (*somewhat homomorphic encryption*). Este esquema é capaz de somar e multiplicar textos cifrados de maneira homomórfica, mas conforme as operações são realizadas, é acrescentado um ruído ao texto cifrado. O algoritmo de decriptação funciona desde que tal ruído não ultrapasse um certo limiar. Usando um conceito que chamou de ***autoinicialização*** (*bootstrapping*), Craig Gentry propõe a construção de um novo esquema, que pode decriptar e reduzir o ruído homomorficamente. Porém, esta adaptação acarreta diretamente no aumento do tamanho dos parâmetros, tornando inviável a implementação do esquema na prática.

1.2. Criptografia completamente homomórfica

Nas próximas seções serão descritos em detalhes as propostas de criptografia completamente homomórfica. Primeiramente, será apresentado o esquema simplificado, que utiliza apenas números inteiros e contém os principais conceitos que também serão utilizados no esquema baseado em reticulados ideais. Ambas as propostas seguem a mesma estratégia, que pode ser resumida de acordo com os seguintes passos:

1. obtenção de um esquema capaz de lidar com uma classe limitada de circuitos, isto é, um esquema de encriptação homomórfica restrita;
2. redução da profundidade do circuito de decriptação;
3. implementação da autoinicialização, permitindo construir um esquema completamente homomórfico em nível.

1.2.1. Segurança

A segurança de um criptossistema contra ***ataque adaptativo de texto cifrado escolhido*** (CCA2 - *chosen-ciphertext attack*) é definida levando em consideração o seguinte jogo:

Configuração. O desafiante obtém $(\text{sk}, \text{pk}) = \text{KeyGen}(\lambda)$ e envia pk para o adversário $\mathscr{A}$.

Consultas. $\mathscr{A}$ envia textos cifrados para o desafiante, antes ou depois do desafio, que retorna o texto claro correspondente.

Desafio. O adversário gera aleatoriamente dois textos claros $m_0, m_1 \in \mathscr{P}$ e manda para o desafiante, que escolhe um bit $b \in \{0,1\}$ aleatoriamente e computa o texto cifrado $c = \text{Enc}(\text{pk}, m_b)$. O desafiante envia c para $\mathscr{A}$.

Resposta. $\mathscr{A}$ manda um bit b' para o desafiante e ganha o jogo se $b' = b$.

O esquema é seguro se não houver um adversário polinomial capaz de vencer este jogo com probabilidade não negligível.

Uma definição ligeiramente diferente, que permite consultas apenas antes de ser feito o desafio, é denotado pela sigla CCA1. Um criptossistema $\mathscr{E}$ é denominado seguro contra ***ataque adaptativo de texto claro escolhido*** se forem permitidas apenas consultas sobre texto claros e não sobre textos cifrados, portanto é um modelo de ataque menos restritivo. Se não for permitida nenhuma consulta, o sistema é denominado ***semanticamente seguro***. Este último modelo, é o mais restritivo, porque garante que o texto cifrado não contém informação a respeito de nenhuma função que possa ser computada eficientemente a partir do texto claro. A segurança com relação a este modelo implica diretamente na impossibilidade de obter um criptossistema que seja capaz de responder consultas de comparação de valores, como por exemplo é necessário para ordenar sequências de valores. Em modelos de computação que utilizam ***ramificação condicional***, isto é, verificam se um valor x é maior ou igual a zero, é possível representar um algoritmo por meio de um programa que contém instruções como laços, saltos condicionais, etc. Esta é uma representação bastante prática em comparação com circuitos algébricos puros (formados apenas por somas e multiplicações), mas infelizmente não é possível obter segurança semântica neste contexto. A possibilidade de verificar se um texto cifrado corresponde a um texto claro cujo valor seja maior que zero, sem conhecimento da chave privada, ou seja, a existência um algoritmo eficiente para computar a função $f : \mathscr{K}_{\text{pub}} \times \mathscr{C} \to \{0,1\}$ tal que $f(\text{pk}, c) = 1$ se e somente se $c = \text{Enc}(\text{pk}, m)$ e $m \geq 0$, é justamente um exemplo de função que contém informação relevante do texto cifrado e fere a definição de segurança semântica.

Duas distribuições são ***indistinguíveis*** caso a adaptação trivial do jogo descrito acima não puder ser ganho por um adversário polinomial.

1.2.2. Homomorfismos secretos

Definição 1.2.1. ***Corretude.*** O esquema $\mathscr{E}(\text{KeyGen}, \text{Dec}, \text{Enc}, \text{Eval})$ é *correto* se, para um determinado circuito C e se para qualquer par de chaves (sk, pk) gerado por KeyGen quaisquer tuplas de mensagens $(m_1, ..., m_t)$ e seus respectivos textos cifrados $\Psi = \langle \psi_1, ..., \psi_t \rangle$, ou seja, $\psi_i = \text{Enc}(\text{pk}, m_i)$ para $1 \leq i \leq t$, então temos que

$$\text{Dec}(\text{sk}, \text{Eval}(\text{pk}, C, \Psi)) = C(m_1, ..., m_t).$$

Além disso, os algoritmos KeyGen, Dec, Enc e Eval devem ter complexidade polinomial.

Encriptação completamente homomórfica. O esquema $\mathscr{E}$ é *correto para uma classe* $\mathbf{S_C}$ de circuitos, se for correto para cada $\mathbf{C} \in \mathbf{S_C}$. Além disso, $\mathscr{E}$ é denomicado *completamente homomórfico* se for correto para todo circuito algébrico. Alternativamente, podemos basear a construção em circuitos booleanos, já que ambos os modelos de computação são equivalentes.

Privacidade do circuito. Dizemos que um esquema $\mathscr{E}$ tem *privacidade de circuito* se as seguintes funções forem indistinguíveis:

$$\text{Enc}(\text{pk}, C(m_1, ..., m_t)) \approx \text{Eval}(\text{pk}, C, \Psi).$$

Encriptação homomórfica compacta. O esquema $\mathscr{E}$ é *compacto* se para todo circuito C, todo conjunto de textos cifrados Ψ, a partir de qualquer chave pública válida, isto é, gerada por KeyGen, então o tamanho do texto cifrado gerado pelo algoritmo Eval é polinomial em relação ao parâmetro de segurança λ e independente do tamanho de C.

Circuito de decriptação aumentado. Seja $\mathscr{E}$ um esquema tal que a decriptação é implementado por um circuito que depende apenas do parâmetro de segurança λ. Define-se o ***conjunto de circuitos de decriptação aumentado*** como sendo o conjunto formado por dois circuitos que recebem como entrada a chave privada e dois textos cifrados. O primeiro circuito, $D_{\mathscr{E}}^{(+)}$, decriptar os textos cifrados de Ψ e soma os resultados, enquanto que o segundo, $D_{\mathscr{E}}^{\times}$, faz o mesmo e ao final multiplica os resultados.

Encriptação com autoinicialização. Seja $\mathscr{E}$ um esquema de encriptação homomórfica. Se $\mathbf{S_C}$ representa o conjunto dos circuitos para o qual $\mathscr{E}$ é correto, e se $D_{\mathscr{E}} \subseteq \mathbf{S_C}$, onde $D_{\mathscr{E}}$ representa o conjunto de circuitos de decriptação aumentado, então $\mathscr{E}$ é denominada ***autoinicializável***.

Encriptação homomórfica em nível. Seja $\mathscr{E}$ um esquema correto para os circuitos de decriptação aumentado, ou seja, $\mathscr{E}$ é ***autoinicializável***, então é possível construir um novo esquema $\mathscr{E}^{(d)}$, correto, compacto e homomórfico para todos os circuitos booleanos de profundidade d. Além disso, $\mathscr{E}^{(d)}$ é semanticamente seguro se $\mathscr{E}$ também é. Especificamente, um ataque de com vantagem ε

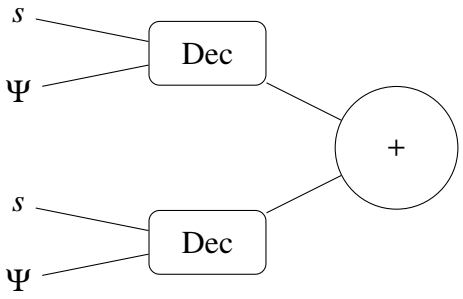

Figura 1.5. $D^{+}_{\mathscr{E}}$

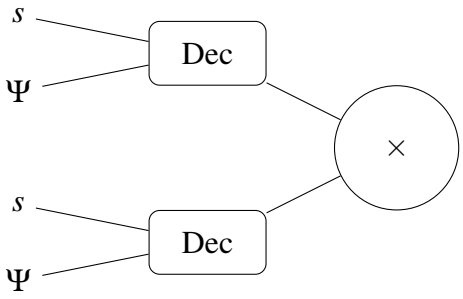

Figura 1.6. $D^{\times}_{\mathscr{E}}$

sobre $\mathscr{E}$ pode ser transformado em um ataque com vantagem $\varepsilon/\ell d$, onde ℓ é o tamanho da chave privada em $\mathscr{E}$.

Este novo esquema utiliza o mesmo $D_{\mathscr{E}}$ e possui mesmo tamanho de chave privada e de texto cifrado. A chave pública consiste de $d+1$ chaves públicas de $\mathscr{E}$, $(\mathrm{pk}_1, ..., \mathrm{pk}_{d+1})$, acrescidas da encriptação de s_i usando pk_{i+1}.

Em cada nível i de um circuito C, os textos são novamente encriptados, utilizando pk_{i+1} e cada soma ou multiplicação do circuito original é substituida por um circuito de decriptação aumentado equivalente. Sendo assim, existe um algoritmo, denotado por Rec, que reencripta a mensagem trocando a chave pública pk_i por pk_{i+1}, de modo que a mensagem sempre está protegida por um nível de encriptação.

Algoritmo 1.2.1 Reencriptação

ENTRADA pk_{i+1}, $D_{\mathscr{E}}$, $\bar{s_i} = \mathrm{Enc}(\mathrm{pk}_{i+1}, s_i)$ e Ψ_i.
SAÍDA um conjunto de textos cifrados usando a chave pk_{i+1} e o conjunto Ψ_i.
$\bar{\Psi}_i = \mathrm{Enc}(\mathrm{pk}_{i+1}, \Psi_i)$.
retorne $\Psi_{i+1} = \mathrm{Eval}(\mathrm{pk}_{i+1}, D_{\mathscr{E}}, (\bar{s_i}, \bar{\Psi}_i))$.

A figura 1.7 mostra um exemplo com um circuito de dois níveis, utilizando quatro mensagens, m_1, m_2, m_3 e m_4, de modo que definimos as seguintes variáveis:

$$\Psi_1 = (\mathrm{Enc}(\mathrm{pk}_1, m_1), \mathrm{Enc}(\mathrm{pk}_1, m_2)),$$
$$\Psi_i^{\mathrm{add}} = \mathrm{Enc}(pk_i, m_1 + m_2),$$
$$\Omega_1 = (\mathrm{Enc}(\mathrm{pk}_1, m_3), \mathrm{Enc}(\mathrm{pk}_1, m_4)),$$
$$\Omega_i^{\mathrm{add}} = \mathrm{Enc}(pk_i, m_3 + m_4).$$

Para simplificar a notação, vamos utilizar $\bar{s_i}$ para denotar o vetor composto pela encriptação de cada bit de s_i usando a chave pública pk_{i+1}. Analogamente, denotamos por $\bar{\Psi}_i$ (ou $\bar{\Omega}_i$) a encriptação de Ψ_i (ou Ω_i) usando a chave pública pk_{i+1}.

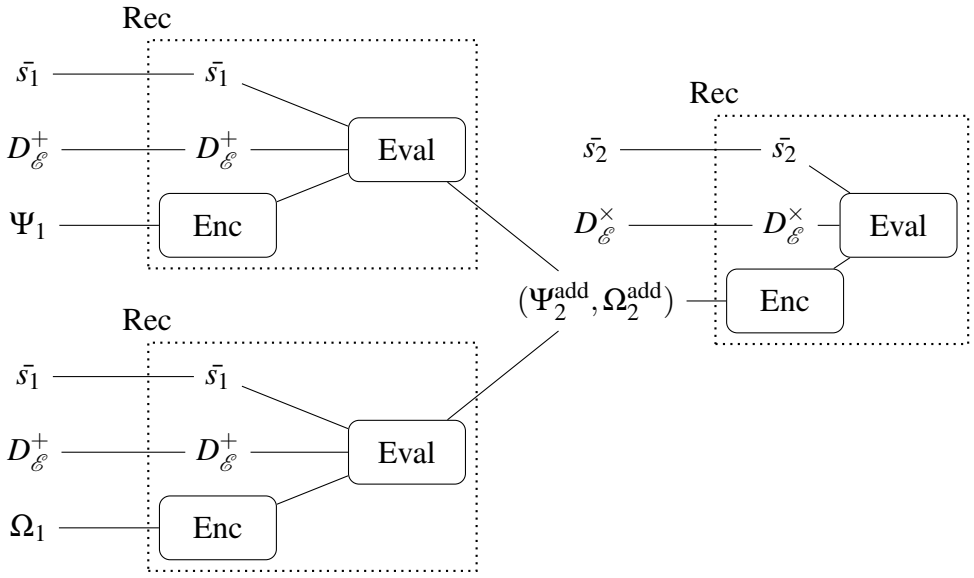

Figura 1.7. $\text{Enc}((m_1+m_2)\times(m_3+m_4))$

Definição 1.2.2. ***Segurança circular.*** Dado um esquema $\mathscr{E}$, dizemos que $\mathscr{E}$ tem segurança circular se for seguro criptografar a chave privada com sua própria chave pública.

Se o esquema $\mathscr{E}$ tiver segurança circular, podemos utilizar a própria chave pública, pk, para encriptar a chave privada sk e portanto não é preciso uma cadeia de $d+1$ chaves públicas.

1.2.3. O esquema sobre os inteiros

Nesta seção será descrito um esquema simplificado, baseado em números inteiros, com a intenção de ilustrar o funcionamento da matemática em dimensão 1. A proposta original de Craig Gentry [Gen09a] estende as mesmas ideias para dimensão n. Será inicialmente descrita a versão simétrica e depois será introduzido o uso do problema SSP para ao mesmo tempo tornar o esquema assimétrico e também para otimizar o algoritmo de decriptação.

1.2.3.1. Versão simétrica

Definição 1.2.3. Seja λ o parâmetro de segurança. O algoritmo KeyGen gera aleatoriamente um inteiro ímpar p com λ^2 bits. Para encriptar um bit m, o algoritmo Enc escolhe m' com λ bits, de modo que m' tenha a mesma paridade de m. É utilizado um inteiro q, com λ^5 bits e o texto cifrado c é calculado da seguinte forma:

$$c = m' + pq.$$

O algoritmo de decriptação computa $m = \mathrm{Dec}(c, p) = \lfloor c \rceil_p \pmod 2$, obtendo de volta o bit encriptado. É simples ver que a encriptação é homomórfica com relação a soma e também com relação a multiplicação. Porém, a decriptação só funciona caso $|m'|$ seja menor que $p/2$, pois a redução módulo p terá a mesma paridade que m.

O problema de encontrar p dados os textos cifrados c_1, c_2, ..., c_k, tal que $c_i = m'_i + pq_i$ e $m'_i \ll p$, denominado ***máximo divisor comum (mdc) aproximado***, foi estudado no contexto de criptoanálise [HG01]. O tamanho de q_i é escolhido para resistir ao ataque descrito neste trabalho.

1.2.3.2. Versão assimétrica

Para tornar o esquema assimétrico a chave privada continua sendo p e a chave pública é formada por encriptações do zero, isto é, inteiros na forma $x_i = 2r_i + pq_i$, para valores de r_i e q_i escolhidos nos mesmos intervalos, de modo que exista um subconjunto cuja soma seja igual a $1/p$. Dada uma mensagem m, o algoritmo de encriptação soma m com um subconjunto aleatório da chave pública. A segurança do esquema passa a depender da dificuldade do problema SSP (*subset sum problem*).

Assim, tendo a solução do problema SSP é possível computar a chave privada, enquanto a chave pública é um conjunto de inteiros que é a entrada do problema SSP. Ou seja, a chave pública é dada por $(s_1, ..., s_k)$ e existe um subconjunto S dos índices tal que $1/p = \sum_{i \in S} s_i$. O algoritmo de encriptação retorna o vetor $(cs_1, ..., cs_k)$ e, para decriptar, calcula-se a soma $\sum_{i \in S} cs_i \pmod 2 = c/p \pmod 2$.

É importante ressaltar que esta ideia possui uma vantagem em relação à decriptação, porque a nova proposta é mais eficiente, já que o cálculo de c/p pode ser efetuado facilmente usando a solução do problema SSP.

Parâmetros. A construção a seguir utiliza diversos parâmetros, cujos tamanhos são polinomiais em relação ao parâmetro de segurança λ:

- γ é o comprimento em bits dos valores de x_i. Este parâmetro deve ser escolhido de maneira tal que $\gamma = \omega(\eta^2 \log \lambda)$, pois assim evita ataques contra o problema do mdc aproximado;

- η é o comprimento em bits da chave secreta p, respeitando a desigualdade $\eta \geq \rho\Theta(\lambda \log^2 \lambda)$, para permitir que o esquema seja capaz de avaliar homomorficamente o circuito reduzido de decriptação;
- ρ é o comprimento em bits do ruído r_i. Este parâmetro deve ser escolhido de forma que $\rho = \omega(log\lambda)$, para que o esquema resista a ataque de força bruta contra o ruído;
- τ é a quantidade de x_i's na chave pública, sendo escolhido de modo que $\tau \geq \gamma + \omega(\log \lambda)$, para que seja possível utilizar o *leftover hash lemma* na redução ao problema do mdc aproximado;
- $\rho' = \rho + \omega(\log \lambda)$ é um parâmetro secundário utilizado no algoritmo de decriptação.

Uma sugestão dada instancia os parâmetros da seguinte forma: $\rho = \lambda$, $\rho' = 2\lambda$, $\eta = \tilde{O}(\lambda^2)$, $\gamma = \tilde{O}(\lambda^5)$ e $\tau = \gamma + \lambda$ [vDGHV09]. Considerando uma escolha de parâmetros como esta, definimos a seguinte distribuição:

$$\mathscr{D}_{\gamma,\rho}(p) = \{x = pq + r \mid q \leftarrow \mathbb{Z} \cap [0, 2^\gamma/p), r \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho)\}.$$

Definição 1.2.4. Geração de chaves. Obtenha um inteiro ímpar p aleatório com η bits. Para $0 \leq i \leq \tau$, compute $x_i = \mathscr{D}_{\gamma,\rho}(p)$. Renomeie os índices de modo que x_0 corresponda ao maior elemento. Faça isso até que x_0 seja ímpar e $x_0 \pmod p$ seja par. A chave pública é dada por $\mathscr{K}_{\text{pub}} = (x_0, ..., x_\tau)$ e a chave privada é dada por $\mathscr{K}_{\text{priv}} = p$.

Encriptação. Escolha um subconjunto aleatório $S \subset \{0, 1, ..., \tau\}$ e um inteiro r aleatório no intervalo $(-2^{\rho'}, 2^{\rho'})$ e compute $c = [m + 2r + \sum_{i \in S} x_i]_{x_0}$.

Decriptação. Retorne $m = [c]_p \pmod 2$.

Avaliação. Dado o circuito C e t textos cifrados, execute as operações de C aos textos cifrados, portanto sobre inteiros grandes, e retorne o valor encontrado.

É importante ressaltar que na medida em que p é ímpar, a decriptação pode ser efetuada da seguinte maneira:

$$m' = [c - \lfloor c/p \rceil]_2 = (c \pmod 2) \oplus (\lfloor c/p \rceil \pmod 2).$$

1.2.3.3. Corretude

Vamos agora demonstrar a corretude do esquema da definição 1.2.4. Contudo, antes disso, serão dadas as definições de circuito generalizado e circuito permitido, que até certo ponto são definições que tornam a demonstração da corretude uma mera aplicação das definições que seguem.

Definição 1.2.5. ***Circuito generalizado.*** Considerando um circuito c cujas portas correspondam a operações de soma e multiplicação módulo 2, o circuito generalizado $g(c)$ é formado por operações equivalentes, porém ao invés de efetuar cálculos sobre bits, computa a soma e multiplicação de números inteiros.

Definição 1.2.6. ***Circuitos permitidos.*** Considerando um circuito c e o circuito generalizado $g(c)$ correspondente, define-se a *classe de circuitos permitidos*, denotada por $C_{\mathscr{E}}$, como sendo aqueles circuitos cujas entradas sejam inteiros valor absoluto no máximo $2^{\alpha(\rho'+2)}$ e cuja saída tenha valor absoluto no máximo $2^{\alpha(\eta-4)}$.

Na verdade, a definição 1.2.6 é uma abordagem que permite demonstrar a corretude do esquema de forma direta, pois os parâmetros foram escolhidos justamente para satisfazê-la. De fato, o ruído máximo de um texto cifrado novo é $2^{\rho'+2}$, enquando que o algoritmo de decriptação espera um texto cifrado com ruído cujo valor absoluto seja limitado por $p/2$, isto é, um valor estritamente menor que $2^{\gamma-2}$. Para permitir a redução do circuito de decriptação, este limite é reduzido para $p/8$, correspondendo portanto ao valor anunciado $2^{\eta-4}$.

Além disso, a partir desta definição não fica claro que tipo de circuito pode ser avaliado. Em especial, o ruído de uma multiplicação é elevado ao quadrado, enquanto que na soma o ruído tem um crescimento linear. Desta forma, interpretando o circuito como um polinômio multivariável, o grau deste polinômio representa a ***profundidade multiplicativa*** do circuito.

Lema 1.2.1. Dado um circuito c e o circuito generalizado $g(c)$ correspondente, construímos o polinômio $f(x_1,...,x_t)$ equivalente a este circuito. Seja d um inteiro correspondente ao grau de f, então se $|f|(2^{\rho'+2})^d \leq 2^{\eta-4}$, onde $|f|$ representa a somatória dos coeficientes de f, temos que c é um circuito permitido, ou seja, $c \in \mathbf{S_C}$.

Prova. De acordo com 1.2.3.2, temos que $c = [m + 2r + \sum_{i\in S} x_i]_{x_0}$. Como x_0 é o valor máximo entre todos os valores de x_i, para $0 \leq i \leq \tau$, então existe um inteiro k, tal que $|k| < \tau$, satisfazendo a seguinte equação

$$c = (m + 2r + \sum_{i\in S} x_i) + kx_0.$$

Pela definição de x_i, temos que $x_i = q_i p + 2r_i$, para $|r_i| \leq 2^{\rho}$. Com isso, temos que

$$c = k(q_0 p + 2r_0) + (m + 2r + \sum_{i\in S}(q_i p + 2r_i)),$$

$$c = p(kq_0 + \sum q_i) + (m + 2r + 2kr_0 + \sum_{i \in S} 2r_i),$$

$$c = p(kq_0 + \sum q_i) + (m + 2(r + kr_0 + \sum_{i \in S} r_i)).$$

Considerando que $\rho' \geq 2\rho$ e $\tau \leq 2^\rho$, já que $\tau = \lambda^5 + \lambda \leq 2^\lambda$ ($\lambda > 23$ é suficiente para garantir essa condição), o termo mais a direita tem valor absoluto no máximo,

$$\begin{aligned} |1 + 2(2^{\rho'+1} + \tau 2^{\rho+1} + \tau 2^{\rho+1})| &\leq |1 + 2(2^{\rho'+1} + \tau 2^{\rho+2})|, \\ &\leq |1 + 2^{\rho'+2} + \tau 2^{\rho+3}|, \\ &\leq |2^{\rho'+3}|. \end{aligned}$$

Portanto, o esquema pode avaliar polinômios cujo grau d respeite a seguinte desigualdade:

$$d \leq \frac{\eta - 4 - \log|f|}{\rho' + 2}.\square$$

Polinômios que satisfaçam essa condição são denominados ***polinômios permitidos.***

Lema 1.2.2. Seja $(\mathrm{sk}, \mathrm{pk})$ um par de chaves gerado por KeyGen. Seja $c = \mathrm{Enc}(\mathrm{pk}, m)$, com $m \in \{0,1\}$. Então, temos que $c \pmod p$ é da forma $2a + m$, ou seja, $c \pmod p$ possui a mesma paridade que m. Além disso, $|2a + m| < 2^{\rho'+2}$.

Prova. De acordo com 1.2.3.2, temos que $c = [m + 2r + \sum_{i \in S} x_i]_{x_0}$. Como x_0 é o valor máximo entre todos os valores de x_i, para $0 \leq i \leq \tau$, então a redução modular de cada x_i por x_0 resulta em um inteiro negativo. Desconsiderando esta parte negativa, sabemos que $2r$ por definição é no máximo $2^{\rho'+2}$. $\square$

Lema 1.2.3. Considerando um circuito permitido $\mathbf{C}$, o resultado de $\mathrm{Eval}(\mathrm{pk}, \mathbf{C}, c_1, ..., c_t)$, onde c_i são textos cifrados válidos, é um texto cifrado cujo rúido é no máximo $p/8$.

Prova. O ruído de $\mathrm{Eval}(\mathrm{pk}, \mathbf{C}, c_1, ..., c_t)$ é dado pela avaliação de $\mathbf{C}$ nos ruídos de c_i, isto é, podemos separar a avaliação do circuito C em duas partes, sendo que a parte múltipla de p resulta em um novo múltiplo de p, enquanto que a avaliação dos ruídos separadamente, resulta no ruído final. Como o ruído de cada c_i é limitado por $2^{\rho'+2}$ de acordo com o lema 1.2.2, então pela definição de circuito permitido, temos que o ruído final é no máximo $2^{\eta-4} = p/8$. $\square$

Infelizmente o esquema $\mathscr{E}$ não possui decriptação com profundidade multiplicativa suficientemente curta para ser completamente homomórfica. Alguns ajustes serão feitos adiante e serão responsáveis pela construção de um novo esquema com circuito de decriptação adequado.

1.2.3.4. Segurança

Os detalhes da demonstração de segurança do esquema da seção anterior podem ser encontrados no trabalho original [vDGHV09], onde mostra-se que a existência de um ataque ao esquema proposto permite resolver o problema do mdc aproximado. Em linhas gerais, o problema é encontrar um divisor comum dentre um conjunto de múltiplos aproximados desse divisor. Supondo a existência de um algoritmo que seja capaz de descobrir um bit do texto claro, é utilizado o algoritmo do mdc binário para construir uma solução para o problema do mdc aproximado.

Neste momento vale apontar o motivo pelo qual foi utilizado o parâmetro secundário ρ'. Basicamente, escolhendo um ruído com $\rho' = 2\rho$ bits, temos que o texto cifrado está protegido por um *ruído alto* ρ', enquanto que a chave pública contém encriptações do zero, realizado com *ruído baixo* ρ. Esta diferença é um ponto chave na redução do criptossistema baseado em ruído alto para o problema do mdc aproximado de ruído baixo.

Definição 1.2.7. O ***problema do mdc aproximado***, parametrizado por (ρ, η, γ), consiste em: dados um número polinomial de elementos da distribuição $\mathscr{D}_{\gamma,\rho}(p)$, para um inteiro ímpar p escolhido aleatoriamente, revele p.

Teorema 1.2.1. Para a escolha de parâmetros realizada na definição 1.2.3.2 e o parâmetro de segurança λ, qualquer ataque $\mathscr{A}$ com vantagem ε sobre o esquema $\mathscr{E}$ pode ser convertido em um algoritmo $\mathscr{B}$ para resolver o problema do mdc aproximado com vantagem pelo menos $\varepsilon/2$. A complexidade de $\mathscr{B}$ é polinomial no tempo de execução de $\mathscr{A}$ e também sobre λ e $1/\varepsilon$.

1.2.3.5. Redução da profundidade do circuito de decriptação

Nesta seção serão apresentadas as ideias utilizadas para reduzir a profundidade do circuito de decriptação. Para construir um esquema completamente homomórfico é preciso que o algoritmo de decriptação possa ser computado por um circuito de profundidade multiplicativa suficientemente baixa. A decriptação é calculada pela expressão $m = [c - [c/p]]_2$, que não parece possuir um circuito com as características desejadas. Para resolver este problema serão acrescentadas ao texto cifrado, informações que ajudam a decriptar a mensagem sem comprometer o esquema. A seguir é apresentado um esquema capaz de avaliar seu próprio circuito de decriptação.

Parâmetros. Essa construção utiliza três novos parâmetros: $\kappa = \gamma\eta/\rho'$, $\theta = \lambda$ e $\Theta = \omega(\kappa \log \lambda)$, ou seja, todos possuem tamanho polinomial no parâmetro de segurança λ.

Geração de chaves. Compute sk e pk como na definição 1.2.3.2. Compute

$x_p = \lfloor 2^\kappa / p \rceil$, escolha aleatoriamente um vetor, $s = \langle s_1, ..., s_\Theta \rangle$, com Θ bits e ***peso de Hamming*** θ. O conjunto S é definido como

$$S = \{i \mid s_i = 1\}.$$

Escolha aleatoriamente inteiros u_i, onde $1 \leq i \leq \Theta$, com no máximo κ bits, tais que $\sum_{i \in S} u_i = x_p \pmod{2^{\kappa+1}}$. Compute $y_i = u_i / 2^\kappa$, de forma que cada y_i é um inteiro positivo menor ou igual a dois, com κ bits de precisão após a vírgula. Assim, temos que $[\sum_{i \in S} y_i]_2 = (1/p) - \Delta_p$, para $\Delta_p < 2^{-\kappa}$.

A chave privada é dada pelo vetor $(s_1, ..., s_\kappa)$ e a chave pública é dada por pk e o vetor $(y_1, ..., y_\Theta)$.

Encriptação. Compute c como no esquema inicial. Para $1 \leq i \leq \Theta$, calcule $z_i = [c y_i]_2$, mantendo apenas $\lceil \log \theta \rceil + 3$ de precisão para cada z_i. Retorne c e o vetor $(z_1, ..., z_\Theta)$.

Decriptação. Retorne $m' = [c - \lfloor \sum_{i \in S} s_i z_i \rceil]_2$.

Avaliação. A soma e multiplicação continuam sendo efetuadas por meio das operações canônicas de números racionais.

Lema 1.2.4. O esquema modificado é correto para circuitos permitidos $C \in \mathbf{S}_{\mathscr{C}}$. Além disso, dado um texto cifrado $(z_1, ..., z_\Theta)$, gerado pela avaliação de um circuito permitido qualquer, temos que $s_i z_i - \lfloor \sum s_i z_i \rceil \leq 1/4$.

Prova. Dado que a chave pública contém o vetor $(y_1, ..., y_\Theta)$, sabe-se que os valores de y_i foram escolhidos de forma que $[\sum s_i y_i]_2 = 1/p + \Delta_p$, onde $\Delta_p \leq 2^{-\kappa}$.

Dado um circuito permitido C, tal que $c^* = \text{Eval}(\text{pk}, C, c_1, ..., c_t)$, para textos cifrados c_i válidos, temos que $[c^* y_i]_2 = z_i - \Delta_i$, com $\Delta_i \leq 1/16\theta$, já que apenas $\lceil \log \theta \rceil + 3$ da precisão é mantida em relação a z_i. Com isso, temos que

$$\begin{aligned}
[(c^*/p) - \sum s_i z_i]_2 &= [(c^*/p) - \sum s_i [c^* y_i]_2 + \sum s_i \Delta_i]_2, \\
&= [(c^*/p) - c^* [\sum s_i y_i]_2 + \sum s_i \Delta_i]_2], \\
&= [(c^*/p) - c^* (1/p - \Delta_p) + \sum s_i \Delta_i]_2, \\
&= [(c^* \Delta_p + \sum s_i \Delta_i]_2.
\end{aligned}$$

Considerando este último termo, temos que $|c^* \Delta_p| \leq 1/16$, pois c^* é um texto cifrado retornado pelo algoritmo de avaliação, cuja entrada é formada por textos cifrados de tamanho no máximo $2^{\alpha(\rho'+2)}$. Assim, o algoritmo Eval retorna um valor com tamanho no máximo $2^{\alpha(\eta-4)}$. Em particular, os textos cifrados são limitados superiormente por 2^γ, de modo que c^* tem magnitude no máximo $2^\gamma(\eta-4)/(\rho'+2) < 2^{\kappa-4}$. Logo, como $\Delta_p < 2^{-\kappa}$, temos que $|c^* \Delta_p| < 1/16$. Já em relação à $|\sum s_i \Delta_i|$, como $|\Delta_i| < 1/16\theta$ e existem θ valores de i para os quais $i \in S$, então temos que $|\sum s_i \Delta_i| < 1/16$. Portanto, temos que

$$|[c^* \Delta_p + \sum s_i \Delta_i]_2| < 1/8. \square$$

1.2.3.6. Autoinicialização

Na seção anterior, além de obter um sistema assimétrico, a ideia fez com que a decriptação seja mais eficiente, permitindo avaliar homomorficamente o próprio circuito de decriptação. Se for possível decriptar o sistema desta forma, reduzindo o ruído, e ainda for possível realizar uma operação extra, de soma ou multiplicação, então conseguiríamos um novo esquema que é capaz de avaliar circuitos de qualquer tamanho.

Até o momento, o esquema descrito permite avaliar circuitos de tamanho limitado, portanto não é completamente homomórfico. Para resolver este problema, Craig Gentry utilizou a ideia que chamou de ***autoinicialização***, construindo uma função que permite reencriptar um texto cifrado de modo a reduzir o ruído. Para fazer isso, é inserida uma dica da chave privada no texto cifrado, com base no ***problema SSP***. Assim, usando um novo par de chaves é possível calcular uma nova encriptação, seguida de uma decriptação com a chave privada original. Com isso, o sistema continua protegido por um nível de encriptação, mas o ruído foi reduzido.

Teorema 1.2.2. Considerando o criptossistema da seção anterior e $D_{\mathscr{E}}$ o conjunto de circuitos de decriptação aumentado, então $D_{\mathscr{E}} \in \mathbf{S_C}$.

Prova. O objetivo é encontrar um circuito adequado para computar a seguinte equação:

$$m = c - \lfloor \sum s_i z_i \rceil \pmod 2.$$

Para auxílio será utilizada uma nova variável $a_i = s_i.z_i$, onde $1 \leq i \leq \Theta$. Com isso, $a_i = z_i$ quando $s_i = 1$ e $a_i = 0$ quando $s_i = 0$. Por definição, a_i possui $n = \lceil \log \theta \rceil + 3$ bits de precisão e existem θ valores de a_i diferentes de zero. Este último fato é crucial para encontrar um circuito adequado, porque permite reduzir a quantidade de variáveis que precisamos lidar. Esta redução é realizada encontrando $n+1 = \lceil \log \theta \rceil + 4$ números racionais w_j, tais que $\sum w_j = \sum a_i \pmod 2$.

Cada a_i é um número racional entre zero e dois. Portanto, a representação binária de a_i pode ser expressa da seguinte maneira:

$$a_i = a_{i,0}, a_{i,-1} a_{i,-2} \ldots a_{i,-n}.$$

Índices negativos são utilizados para reforçar o fato de que estes bits representam a expansão binária do número racional, ou seja, $a_i = 2^{-j} \sum_{j=0}^{n} a_{i,-j}$.

Antes de calcular w_j, vamos definir W_{-j} como sendo o ***peso de Hamming*** do vetor $\{a_{i,j}\}_{i=1}^{\theta}$, como mostra a tabela 1.1 a seguir:

Como não há mais que θ valores de a_i não nulos, então o valor de W_{-j} é no máximo θ e definindo $w_j = 2^{-j} W_{-j} \pmod 2$, temos que w_j pode ser representado por $\lceil \log \theta \rceil + 1$ bits de precisão.

$a_{1,0},$	$a_{1,-1}$	$a_{1,-2}$	...	$a_{1,-n}$
$a_{2,0},$	$a_{2,-1}$	$a_{2,-2}$	...	$a_{2,-n}$
$a_{3,0},$	$a_{3,-1}$	$a_{3,-2}$	...	$a_{3,-n}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
$a_{\theta,0},$	$a_{\theta,-1}$	$a_{\theta,-2}$	...	$a_{\theta,-n}$
W_0	W_{-1}	W_{-2}	...	W_{-n}

Tabela 1.1.

Lema 1.2.5. Considerando a sequência de bits $\overrightarrow{b} = (b_1, ..., b_k)$, o ***peso de Hamming*** de $\overrightarrow{b}$, denotado por $H_{\overrightarrow{b}}$ pode ser computado calculando cada um de seus bits. Se a representação binária de $H_{\overrightarrow{b}}$ for dada por $(h_n, ..., h_0)$, de modo que $H_{\overrightarrow{b}} = \sum 2^i h_i$, então h_i pode ser expresso por um polinômio de grau 2^i nas variáveis $\{b_i\}_1^k$. Além disso, existe um circuito de tamanho $k2^i$ que computa todos os valores de h_i simultaneamente.

Prova. O i-ésimo bit de $H_{\overrightarrow{b}}$ pode ser computado por δ_{2^i}, onde δ_i representa o i-ésimo ***polinômio simétrico elementar***. O grau de δ_{2^i} é exatamente 2^i e para calcular simultaneamente todos os valores de h_i basta computar o polinômio $p(z) = \prod(z - b_i)$, já que h_i corresponde ao coeficiente do termo z^{k-i} em $p(z)$.

O algoritmo a seguir computa os bits $h_0, ..., h_n$ e pode ser facilmente transformado em um circuito:

Algoritmo 1.2.2 Polinômios simétricos elementares

ENTRADA $b_1, ..., b_k$.
SAÍDA $\delta_1(b_1, ..., b_k), ..., \delta_n(b_1, ..., b_k)$.
  Inicialize $e_{0,0} = 1$ e $e_{i,0} = 0$ para $i = 1, 2, 3, ..., 2^n$.
  para $j = 1, 2, ..., k$ **faça**
    para $i = 2^\ell, 2^{\ell-1}, ..., 1$ **faça**
      Compute $e_{i,j} b_j e_{i-1,j-1} + e_{i,j-1}$ (aritmética polinomial).
  retorne $e_{1,k}, ..., e_{2^n,k}$.

As multiplicações de polinômios são realizadas com o auxílio da ***transformada rápida de Fourier (FFT)***. □

1.2.3.7. Segurança do novo esquema

Para que a nova proposta seja segura é preciso garantir que as informações inclusas na chave pública, ou seja, $(y_1, ..., y_\theta)$, não possam ser usadas para reconstruir a chave privada. Este problema foi considerado por Craig Gentry em 2009 [Gen09a] e é conhecido como ***problema da soma em subconjunto esparço*** (SSSP - *sparse subset sum problem*). Para que este problema seja difícil é preciso escolher θ

suficientemente grande para evitar ataques de força bruta. Além disso, é necessário ter Θ maior que $\omega(\kappa \log \lambda)$, onde κ é o comprimento em bits dos números incluídos na chave pública.

1.3. O esquema sobre reticulados ideais

1.3.1. Introdução

Na seção anterior foi descrito um esquema baseado em números inteiros, com a intenção de simplificar os conceitos que serão necessários para descrever o esquema sobre reticulados. Existe portanto uma correspondência direta entre as ideias apresentadas nesta seção com a seção anterior, porque o caso sobre números inteiros é um caso particular da construção que será descrita nesta seção. Assim, o leitor pode esperar um grau de abstração maior, embora seja seguida a mesma cadeia de definições e teoremas até atingir o resultado desejado: um esquema capaz de avaliar o seu próprio circuito de decriptação acrescido de uma soma ou multiplicação. Com isso, será possível construir novamente um esquema que possua ***encriptação homomórfica em nível***.

Mas tendo em vista as dificuldades encontradas já na versão baseada em números inteiros, vamos utilizar definições mais abstratas para construir um esquema inicial, para depois concretizar a construção utilizando reticulados ideais, assim como foi feito em [Gen09b].

As definições da seção 1.2 serão utilizadas como base da construção aqui apresentada, portanto é pré-requisito para o entendimento do esquema que será descrito. Em particular, a estratégia de encontrar um esquema inicial, capaz de avaliar uma classe limitada de circuitos, para depois reduzir a profundidade do ***circuito de decriptação*** e, finalmente, utilizar a ***autoinicialização*** para tornar o esquema completamente homomórfico, será novamente o eixo principal que será seguido.

1.3.2. Resumo

As mesmas ideias que funcionam sobre os números inteiros podem ser usadas com anéis polinomiais ideais, onde os ideais I e J são utilizados com a mesma função dos inteiros (2) e (q). Isto é, um vetor m é encriptado computando $c = m + i + j$ e para decriptar, calcula-se

$$m = (c \pmod{B_I}) \pmod{B_J}.$$

Na prática, temos que $B_I = (2)$ e $B_J = (a(x))$, onde $a(x)$ tem grau suficientemente grande para que o espaço de busca por força bruta tenha tamanho λ e além disso, seja possível efetuar pelo menos uma operação de multiplicação de forma homomórfica.

Observando por um outro ângulo, a decriptação está relacionada ao problema do vetor mais próximo em reticulados, já que $\lfloor c \rceil_{B_J}$ é uma instância do ***problema CVP***. Assim, para que o esquema seja seguro, B_J deve ser suficiente-

mente não ortonormal, para que não seja possível usar o algoritmo de Babai para decriptar a mensagem.

Portanto, o esquema de Gentry é semelhante ao criptossistema GGH, já que utiliza uma base boa B_J^{sk}, gerada por um polinômio com coeficientes pequenos. Já a chave pública B_J^{pk} é calculada usando a forma normal de Hermite desta primeira base. Com isso, a demonstração de segurança do esquema é baseada na complexidade dos problemas difíceis em reticulados.

1.3.3. Esquema Abstrato

O esquema abstrato será descrito em termos de anéis e ideais. Sendo assim, considere um anel genérico R gerado de acordo com o parâmetro de segurança λ. Seja $I \subset R$ um ideal e B_I uma base de I.

Dados R e B_I, o algoritmo $\text{IdealGen}(R, B_I)$ retorna as bases pública e privada B_J^{pk} e B_J^{sk}, respectivamente, onde J é o ideal gerado independentemente por B_J^{pk} ou B_J^{sk}, tal que $I + J = R$, isto é, J é relativamente primo a I.

Dado um anel R, uma base $B_I \subset R$ e um elemento $r \in R$, então a notação $r \pmod{B_I}$ é utilizada para descrever o elemento representativo único $r^* \in R$ tal que $r * -r \in I$, isto é, dada a classe lateral $r + I$, existe um único elemento que a representa e este elemento pode ser diferente de acordo com a base escolhida. A notação $R \pmod{B_I}$ remete ao conjunto de elementos representativos distintos com respeito à base B_I.

Dado um anel R, as bases B_I e B_J dos ideais I e J, e um elemento $r \in R$, definimos o algoritmo $\mathscr{D}_{B_I,B_J}(r)$ simplesmente como uma forma de extrair aleatoriamente um elemento da classe lateral $r + I$.

Analogamente à definição 1.2.5, dado um circuito C composto de operações módulo B_I (espaço de texto claro), define-se o *circuito generalizado* como sendo o circuito construído a partir de C trocando-se as operações módulo B_I por operações equivalentes no anel R.

Seja X_{enc} a imagem de $\mathscr{D}_{B_I,B_J}$, de modo que todo texto cifrado é da forma $X_{\text{enc}} + J$. Além disso, seja $X_{\text{dec}} = R \pmod{B_J^{\text{sk}}}$, isto é, os elementos representativos das classes laterais de J com relação a base B_J^{sk}, de maneira que o algoritmo Dec só é capaz de decriptar textos cifrados pertencentes a X_{dec}. Assim como foi definido em 1.2.6, definimos o conjunto de *circuitos permitidos* como sendo

$$\mathscr{C}_{\mathscr{E}}^* = \{C \mid \forall (x_1, ..., x_t) \in X_{\text{enc}}^t, g(C)(x_1, ..., x_t) \in X_{\text{dec}}\}.$$

Mas diferentemente da definição 1.2.6, onde foram estabelecidos valores concretos para X_{enc} e X_{dec}, será utilizada esta versão mais abstrata da definição, permitindo demonstrar de forma simples a corretude do esquema abstrato.

A seguir definimos o esquema abstrato $\mathscr{E}(\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Eval})$, onde o espaço de texto claro $\mathscr{P}$ é dado por $R \pmod{B_I}$ e o algoritmo Eval recebe como parâmetro um circuito cujas portas correspondem a operações realizadas módulo

B_I.

Geração de chaves. O algoritmo KeyGen recebe como parâmetros o anel R e a base B_I e executa o algoritmo IdealGen(R, B_I) para obter as bases B_J^{pk} e B_J^{sk}. A chave pública corresponde a pk $= \{R, B_I, B_J^{\text{pk}}, \mathscr{D}_{B_I,B_J}\}$, enquanto a chave privada é dada por sk $= \{B_J^{\text{sk}}\}$.

Encriptação. Dada a chave pública pk e um texto claro $m \in \mathscr{P}$, o algoritmo Enc(pk, m) retorna $c = m + \mathscr{D}_{B_I,B_J} \pmod{B_J^{\text{pk}}}$.

Decriptação. Dada a chave privada B_J^{sk} e o texto cifrado c, o algoritmo Dec(B_J^{sk}, c) retorna $m = (c \pmod{B_J^{\text{sk}}}) \pmod{B_I}$.

Avaliação. Dada a chave pública B_J^{pk}, um circuito permitido $C \in \mathscr{C}_{\mathscr{E}}$ e um conjunto de textos cifrados $\Psi = (\psi_1, ..., \psi_t)$, o algoritmo Eval$(B_J^{\text{pk}}, C, \Psi)$ executa as operações Add e Mult do circuito generalizado $g(C)$ e retorne o texto cifrado $\psi = g(C)(\Psi)$.

Teorema 1.3.1. O esquema abstrato $\mathscr{E}$ é correto para circuitos permitidos $C \in \mathbf{S_C}$.

Prova. Para qualquer $\Psi = (\psi_1, ..., \psi_t)$, tal que $\psi_= m_k + i_k + j_k$, onde $i_k \in I$ e $j_k \in J$, ou seja, um conjunto de textos cifrados válidos, temos que

$$\begin{aligned} \text{Eval}(\text{pk}, C, \Psi) &= g(C)(\Psi) \pmod{B_J^{\text{pk}}}, \\ &\in g(C)(m_1 + i_1, ..., m_t + i_t) + J. \end{aligned}$$

Como $m_k + i_k \in X_{\text{enc}}$, temos que $g(C)(m_1 + i_1, ..., m_t + i_t) \in X_{\text{dec}}$, por definição. Logo,

$$\begin{aligned} \text{Dec}(\text{sk}, \text{Eval}(\text{pk}, C, \Psi) &= g(C)(m_1 + i_1, ..., m_t + i_t) + J, \\ &= g(C)(m_1 + i_1, ..., m_t + i_t) + \pmod{B_I}, \\ &= g(C)(m_1, ..., m_t) + \pmod{B_I}, \\ &= C(m_1, ..., m_t). \square \end{aligned}$$

1.3.4. Segurança do esquema abstrato

Definição 1.3.1. Problema da classe lateral em ideais (*ideal coset problem - ICP*). Dado um anel R, uma base B_I e os algoritmos IdealGen e $\mathscr{D}_{B_I,B_J}$, que retorna um elemento aleatório de R, o desafiante escolhe aleatoriamente um bit $b \in \{0, 1\}$, gera $(B_J^{\text{sk}}, B_J^{\text{pk}}) = \text{IdealGen}(R, B_I)$. Se $b = 0$, ele computa $r = \mathscr{D}_{B_I,B_J}$ e $t = r \pmod{B)_J^{\text{pk}}}$. Se $b = 1$, ele escolhe t uniformemente em $R \pmod{B)_J^{\text{pk}}}$. O problema consiste em encontrar b dados (t, B_J^{pk}).

Resumidamente, o problema é distinguir entre uma distribuição uniforme e uma distribuição especial, induzida por $\mathscr{D}_{B_I,B_J}$.

Teorema 1.3.2. Suponha a existência de um algoritmo $\mathscr{A}$ capaz de atacar o esquema abstrato $\mathscr{E}$ com vantagem ε. Então existe um algoritmo $\mathscr{B}$, com tempo de execução polinomial em função do tempo de execução de $\mathscr{A}$, que resolve o problema ICP com vantagem $\varepsilon/2$.

Prova. O desafiante manda uma instância (t, B_J^{pk}) do problema ICP para o algoritmo $\mathscr{B}$, que escolhe $s \in I$. $\mathscr{A}$ solicita um desafio sobre o par de textos claros $(m_0, m_1) \in \mathscr{P}$, $\mathscr{B}$ escolhe aleatoriamente o bit $b \in \{0,1\}$ e devolve para $\mathscr{A}$ o valor $\psi = m_b + ts \pmod{B}_J^{\text{pk}}$. O algoritmo $\mathscr{A}$ retorna o palpite β' e $\mathscr{B}$ computa seu próprio palpite $b' = \beta \oplus \beta'$.

Se $b = 0$, ψ corresponde a um texto cifrado válido, pois temos que $\psi = m_b + rs \pmod{B_J^{\text{pk}}}$. Com isso, o algoritmo $\mathscr{A}$ tem vantagem ε. Se $b = 1$, t é uniforme módulo J. Como o ideal gerado por s é primo com J, ts é uniforme módulo J e consequentemente ψ é um elemento aleatoriamente uniforme em $R \pmod{B}_J^{\text{pk}}$, portanto é independente de β, de forma que $\mathscr{A}$ tem vantagem zero. Juntando ambas as possibilidades, temos que a vantagem de $\mathscr{B}$ é $\varepsilon/2$. □

1.3.5. Ideais em anéis polinomiais e reticulados ideais

Considere o anel polinomial $R = \mathbb{Z}[x]/f(x)$, onde $f(x)$ é um polinômio irredutível de grau N sobre $\mathbb{Z}[x]$. Seja $a(x) \in \mathbb{Z}[x]/f(x)$, o ideal gerado por $a(x)$ é formado por todos os polinômios múltiplos de $a(x)$ módulo $f(x)$. Este ideal pode ser representado por um reticulado $\mathscr{L}_R$, onde a base deste reticulado é dada pelos vetores gerados pelos coeficientes dos polinômios $a(x) \pmod{f(x)}$, $x.a(x) \pmod{f(x)}$, $x^2.a(x) \pmod{f(x)}$, ..., $x^{N-1}.a(x) \pmod{f(x)}$, denominada *base de rotação*. Estes vetores são linearmente independentes no espaço vetorial com a base canônica. Em geral, não é necessário que o ideal seja gerado por apenas um polinômio, assim como também não é obrigatório utilizar uma base de rotação.

Dado um ideal sobre um anel polinomial, é possível determinar um reticulado tal que todo ponto do reticulado corresponde a um polinômio do ideal. Este reticulado é denominado ***reticulado ideal***.

A ***forma normal de Hermite*** (*Hermite normal form - HNF*) de um reticulado $\mathscr{L}_R$ é uma ***base triangular superior***, que pode ser computada eficientemente a partir de uma base qualquer do mesmo reticulado, sendo apropriada para ser utilizada como chave pública.

Reticulados ideais são apropriados para concretizarem o esquema abstrato discutido anteriormente, porque a operação de redução modular por uma base B_I é facilmente computada como o elemento pertencente ao paralelepípedo fundamental centralizado $\mathscr{P}(B_I)$. Dado $t \in R$, a redução modular $t \pmod{B_I}$ é computada da seguinte forma

$$t - B_I.\lfloor B_I^{-1} t \rceil.$$

1.3.6. O esquema concreto

Nesta seção será apresentado o esquema $\mathscr{E}$ que concretiza a proposta abstrata discutida anteriormente, utilizando reticulados ideais.

Definição 1.3.2. Seja r_{enc} o menor valor tal que $X_{\text{enc}} \in \mathscr{B}(r_{\text{enc}})$, onde $\mathscr{B}(r)$ é a esfera de raio r. Analogamente, r_{dec} é definido como o menor valor tal que $X_{\text{dec}} \in \mathscr{B}(r_{\text{dec}})$.

Com isso, o conjunto de circuitos permitidos é dado por

$$\mathbf{S}_{\mathbf{C}} = \{\mathbf{C} \mid \forall (x_1,...,x_t) \in \mathscr{B}(r_{\text{enc}})^t, g(\mathbf{C})(x_1,...,x_t) \in \mathscr{B}(r_{\text{dec}})\}.$$

Comparando com o esquema abstrato, X_{enc} e X_{dec} foram substituídos por $\mathscr{B}(r_{\text{enc}})$ e $\mathscr{B}(r_{\text{dec}})$, respectivamente.

Para compreender que classe de circuitos $\mathscr{E}$ consegue avaliar, é preciso estabelecer limites para o tamanho dos vetores resultantes da soma e multiplicação de quaisquer dois vetores. Dados u e v, pela desigualdade triangular, temos que $||u+v|| \leq ||u|| + ||v||$. Já a multiplicação depende do anel R para que possamos estabelecer tal limite. Sendo assim, dizemos que $||u.v|| \leq \gamma_{\text{Mult}}(R).||u||.||v||$, onde $\gamma_{\text{Mult}}(R)$ é um fator dependente de R.

Teorema 1.3.3. Suponha que $r_{\text{enc}} \geq 1$. Dado um circuito $\mathbf{C}$, cujas portas aditivas possuam γ_{Mult} parâmetros de entrada e cujas portas multiplicativas possuam dois parâmetros de entrada, se a profundidade de $\mathbf{C}$ é no máximo $\log\log r_{\text{dec}} - \log\log \gamma_{\text{Mult}}.r_{\text{enc}}$, então $\mathbf{C}(x_1,...,x_t) \in \mathscr{B}(r_{\text{dec}})$, para todo $(x_1,...,x_t) \in \mathscr{B}(r_{\text{enc}})$.

Prova. Considere um circuito $\mathbf{C}$ com profundidade d. Seja r_i um limite superior para a norma dos valores de $\mathbf{C}$ no nível i, onde o nível d representa a entrada do circuito e r_0 representa a sua saída. Uma porta aditiva no nível i gera uma saída $v_+ \in R$, tal que $||v_+|| \leq \gamma_{\text{Mult}}.r_i$, enquanto uma porta multiplicativa no mesmo nível, gera uma saída $v_\times \in R$, tal que $||v_\times|| \leq \gamma_{\text{Mult}}.r_i^2$. Portanto, na pior das hipóteses temos que $r_{i-1} \leq \gamma_{\text{Mult}}.r_i^2$. Dado que $\mathbf{C}$ recebe textos cifrados válidos como entrada, temos que $r_d \leq r_{\text{enc}}$, e com isso, obtemos

$$r_0 \leq (\gamma_{\text{Mult}}.r_{\text{enc}})^{2^d}.\square$$

Portanto, para maximizar a profundidade de circuito que o esquema $\mathscr{E}$ é capaz de lidar homomorficamente, é preciso minimizar γ_{Mult} e r_{enc}. Por outro lado, é preciso maximizar r_{dec}. Porém, a segurança semântica de $\mathscr{E}$ está relacionada à razão $r_{\text{dec}}/r_{\text{enc}}$ [Gen09b]. Para que o esquema seja capaz de decriptar, é necessário que $r_{\text{dec}} < \lambda_1(J)$, e para que o algoritmo ***LLL*** não possa ser usado para atacar o esquema, é necessário que $\lambda_1(J)/r_{\text{enc}}$ não seja muito grande. Isto é, $r_{\text{dec}} = 2^{n_1^c}$ e $\lambda_1(J)/r_{\text{enc}} = 2^{n_2^c}$, para $0 < c_1, c_2 < 1$ é uma escolha que permite avaliar circuitos de profundidade $(c_1 - c_2)\log n$.

Lema 1.3.1. Seja B uma base de um determinado reticulado e $B^* = (B^{-1})^T$. Seja r o raio da maior esfera centrada na origem tal que $\mathscr{P}(B)$ a circunscreve. Então $r = 1/(2.||B^*||)$. Em particular,

$$r_{\text{dec}} = 1/(2.||((B_J^{\text{sk}})^{-1})^T||).$$

Suponha que $||t|| \leq r$, então cada coeficiente de $B^{-1}t$ tem magnitude no máximo $1/2$.

Prova. Cada coeficiente de $B^{-1}t$ é o produto interno de t e uma coluna de B^*, portanto tem magnitude no máximo $||t||.||B^*|| < 1/2$, o que implica que $\lfloor B^{-1}t \rceil = 0$. Assim, $t = t \pmod{B}$ e portanto $t \in \mathscr{B}$. Seja v o maior vetor de B^* e seja x um vetor paralelo a v. Então, se o produto interno entre v e x for estritamente maior que $1/2$, temos que $x \notin \mathscr{P}(B)$, se e somente se $||x|| > 1/(2||B^*||)$. □

O algoritmo IdealGen pode computar B_J^{sk} por meio da base de rotação de um vetor v pequeno e paralelo a $e_1 = (1,0,...,0)$ e $B_J^{\text{pk}} = HNF(B_J^{\text{sk}})$. Com isso, obtemos $r_{\text{dec}} = ||v||/2$.

1.3.7. Redução do circuito de decriptação

Nesta seção serão introduzidos dois ajustes que serão responsáveis por tornar o circuito de decriptação do esquema capaz de ser avaliado homomorficamente.

Ajuste 1. Redefina o conjunto de circuitos permitidos $\mathbf{S_C}$ substituindo $\mathscr{B}(r_{\text{dec}})$ por $\mathscr{B}(r_{\text{dec}})/2$.

Lema 1.3.2. Após o ajuste 1, os coeficientes de $(B_J^{\text{sk}})^{-1}\psi$ distam $1/4$ de um inteiro, onde ψ é um texto cifrado válido.

Ajuste 2. Compute um vetor pequeno $v_J^{\text{sk}} \in J^{-1}$, tal que existe $u \in I+1$ e $u(v_J^{\text{sk}})^{-1} \in I+1$. Além disso, modifique $\mathbf{S_C}$ de modo a usar o seguinte limite para a decriptação

$$\mathscr{B}(2r_{\text{dec}}/(n^{1.5}\gamma_{\text{Mult}}(R)^2||B_I||)).$$

Com este segundo ajuste, a decriptação pode ser realizada da seguinte forma:

$$\psi - B_J^{\text{sk}}.\lfloor (B_J^{\text{sk}})^{-1}\psi \rceil \pmod{B_I} = \psi - \lfloor v_J^{\text{sk}}\psi \rceil \pmod{B_I}.$$

Basicamente a mesma estratégia que foi adotada com números inteiros será também seguida com reticulados ideais. Ou seja, o problema SSSP é introduzido de modo que o texto cifrado passa a conter uma sequência de valores, que ao serem somados permitem a decriptação da mensagem. Para calcular a somatória é utilizada a mesma técnica baseada em polinômios simétricos elementares para computar os bits do ***peso de Hamming***. Porém, o espaço de texto claro $\mathscr{P}$ precisa

ser restrito a $\{0,1\}$ e o ideal I deve ser simplesmente $(2.e_1)$ para obter o resultado desejado.

São definidos dois novos algoritmos para formalizar as ideias anteriormente descritas: SplitKey e ExpandCT. O primeiro é responsável por modificar o par de chaves original, acrescentando uma instância do problema SSSP, de modo que a nova chave privada passa a conter os índices i dos elementos t_i que devem ser somados para obter o vetor v_J^{sk}, assim como ocorreu anteriormente para esconder $1/p$. O segundo modifica o texto cifrado retornado pelos algoritmos Enc e Eval, de modo que, dado ψ válido, o novo texto cifrado é uma sequência de valores $t_i\psi$. Isto é, parte da computação do algoritmo Dec já é realizada pelo algoritmo ExpandCT, restando apenas calcular a somatória dos valores onde i corresponde a um índice válido para a solução do ***problema SSSP***.

Modificação da chave. Sejam as funções $\gamma_{\text{set}}(n)$, simultaneamente pertencente a $\omega(n)$ e a $\text{poly}(n)$ (correspondente a Θ) e $\gamma_{\text{subset}}(n)$, tais que $\gamma_{\text{subset}}(n)$ é ao mesmo tempo $\omega(1)$ e $o(n)$ (correspondente a θ). O algoritmo SplitKey gera $\gamma_{\text{set}}(n)$ valores de t_i em $J^{-1} \pmod{B_I}$, de modo que exista um conjunto T, composto de $\gamma_{\text{subset}}(n)$ valores de t_i, que somados resultam em $v_J^{\text{sk}} + I$. Os índices são representados por um conjunto de bits $\{s_i\}_0^{\gamma_{\text{set}}(n)}$, onde $s_i = 1$ se e somente se t_i pertence a T. A chave pública é modificada para incluir os valores de t_i, enquanto a chave privada é alterada para conter os índices i da solução do problema SSSP.

Expansão do texto cifrado. Dado ψ um texto cifrado válido segundo o esquema $\mathscr{E}$ original, então o algoritmo ExpandCT retorna $c_i = t_i\psi \pmod{B_I}$.

1.3.8. Autoinicialização

Considere a existência de um esquema criptográfico $\mathscr{E}$ capaz de avaliar compactamente uma classe $\mathbf{S_C}$ de circuitos. Em outras palavras, dado um circuito $\mathbf{C} \in \mathbf{S_C}$, o esquema $\mathscr{E}$ é homomórfico em relação à $\mathbf{C}$, portanto existe um circuito $\mathbf{C}'$, estruturalmente idêntico a $\mathbf{C}$, mas que aceita como entrada $\text{Enc}(\text{pk}, m)$ ao invés de m. Então é possível utilizar $\mathscr{E}$ para obter um novo criptossistema $\mathscr{E}'$, capaz de avaliar circuitos de profundidade arbitrária. Para que isso seja possível é necessário que $\mathscr{E}$ seja capaz de avaliar seu próprio circuito de decriptação, acrescentado de uma operação de soma ou multiplicação.

Utilizando d para denotar a profundidade do circuito, representamos por $\mathscr{E}^{(d)}$ o esquema criptográfico capaz de avaliar compactamente circuitos de profundidade no máximo d. O esquema $\mathscr{E}$ pode ser usado para construir $\mathscr{E}^{(d)}$ repetindo o procedimento descrito no parágrafo anterior d vezes.

Mas até o momento ainda não foi descrito o circuito de decriptação que será utilizado. Seguindo a estratégia da seção 1.2.3.6, define-se a variável $a_i = s_i.c_i$, isto é, aqueles valores de c_i correspondentes a uma solução do problema SSSP. Define-se também um conjunto de valores $\{w_i\}_0^{\gamma_{\text{subset}}(n)+1}$ cuja soma, após tomado o inteiro mais próximo de cada coordenada, seja igual a soma dos valores de a_i. Assim, é possível utilizar o algoritmo 1.2.3.6 para computar eficientemente o circuito de decriptação.

De acordo com a escolha de parâmetros, atacar o problema SSSP tem complexidade $2^{\gamma_{\text{subset}}(n)}$. Porém, quanto maior é o valor de $\gamma_{\text{subset}}(n)$, maior é o fator de aproximação para o ***problema CVP***. Considerando que um fator de aproximação de 2^k leva tempo $2^{n/k}$, é preciso escolher $\gamma_{\text{subset}}(n)$ de maneira que ambos os problemas sejam difíceis de atacar. Ou seja, se $\gamma_{\text{subset}}(n) = \sqrt{n}$, a complexidade do problema é aproximadamente $2^{\sqrt{n}}$. Assim, n é escolhido de forma que $n \approx \lambda^2$. Utilizando FFT, é possível obter complexidade aproximadamente de λ^6 para o algoritmo de decriptação [Gen09b]. Em relação ao esquema definido sobre inteiros, o tamanho do parâmetro $q \approx \lambda^5$, para garantir que o problema do mdc aproximado seja difícil, é um fator que torna o esquema baseado em inteiros menos eficiente que a versão baseada em reticulados ideais.

1.4. Trabalhos recentes

Nesta seção será apresentada uma compilação de trabalhos recentes, que propõe a utilização do problema LWE polinomial para obter um esquema homomórfico restrito, isto é, capaz de avaliar uma classe limitada de circuitos algébricos. Além disso, será discutido o uso prático deste tipo de criptossistema, já que muitas aplicações interessantes não requerem a existência de um esquema completamente homomórfico.

1.4.1. ECH sem *autoinicialização*

Até o momento, todas as propostas de ECH apresentadas seguem o modelo desenvolvido por Craig Gentry, onde primeiramente é construído um esquema homomórfico restrito, seguido de uma redução do circuito de decriptação, para finalmente utilizar a ***autoinicialização***. Porém, este modelo possui limites claros em relação à performance mínima que pode ser obtida. Em um trabalho recente são apresentados argumentos que mostram que a *autoinicialização* tem complexidade mínima de $\Omega(\lambda^4)$ [BGV11]. Stehlé e Steinfeld propuseram uma otimização (não tem sido utilizada) que permite reduzir o grau da decriptação para $O(\sqrt{\lambda})$, de modo que a complexidade mínima da ***autoinicialização*** pode tornar-se $\Omega(\lambda^{3.5})$.

Recentemente, em dois trabalhos distintos, Gentry e Halevi [GH11a] e Brakerski e Vaikuntanathan [BV11] encontraram formas de desviar deste modelo principal. No primeiro trabalho, o circuito de decriptação é descrito por polinômios simétricos, cuja computação pode ser realizada por um circuito de profundidade 3, onde o primeiro e terceiro níveis são constituídos apenas de somas, enquanto o segundo nível é constituído de multiplicações. Assim, para evitar o aumento quadrático do ruído, as multiplicações são realizadas utilizando um criptossistema como o ElGamal, capaz de multiplicar homomorficamente. No segundo trabalho, são utilizadas duas técnicas: ***redução de dimensão*** e ***redução de módulo***. Para isso, o esquema é baseado no problema LWE, introduzindo uma importante mudança na construção de ECH eficiente.

Mas apesar das novas ideias, a performance ainda era limitada inferiormente por $\Omega(\lambda^4)$. Em outro trabalho [BGV11], Gentry, Brakerski e Vaikun-

tanathan utilizam algumas das ideias anteriores e o problema LWE em anéis (RLWE) para obter um esquema que não precisa de autoinicialização.

Definição 1.4.1. O ***problema LWE*** consiste em encontrar o vetor $s \in \mathbb{Z}_q^n$, dadas as equações

$$\begin{array}{rcl} \langle s, a_1 \rangle & \approx_{\mathscr{D}} & b_1 \pmod{q} \\ \langle s, a_2 \rangle & \approx_{\mathscr{D}} & b_2 \pmod{q} \\ & \vdots & \end{array}$$

A notação $\approx_{\mathscr{D}}$ significa uma tolerância na igualdade, de acordo com a distribuição $\mathscr{D}$. Ou seja, $\langle s, a_i \rangle$ difere de b_i e esta diferença é determinada pela distribuição $\mathscr{D}$, geralmente tomada como sendo a distribuição normal. Alternativamente, podemos escrever $\langle s, a_i \rangle = b_i + e_i$, onde $e_i \in \mathscr{D}$.

A quantidade de equações contribui relativamente pouco para a solução do problema. Existe um compromisso entre o número de equações e o tempo de execução para encontrar a solução do problema, mesmo com uma quantidade arbitrária de equações a complexidade é na melhor das hipóteses subexponencial [BKW03].

Em 2005, Oded Regev apresenta uma redução quântica do problema LWE ao pior caso de problemas em reticulados. Além disso, este trabalho mostra um novo criptossistema, cuja performance é consideralvelmente melhor que outros esquemas baseados em reticulados [Reg05].

Lyubaskevsky, Peikert e Regev definiram uma versão similar ao problema LWE, mas usando anéis polinomiais [LPR10]. Seja $f(x) = x^d + 1$, onde d é uma potência de 2. Dado um inteiro q e um elemento $s \in R = \mathbb{Z}_q[x]/f(x)$, o **problema LWE em anel** sobre R, com relação a uma distribuição $\mathscr{D}$, é definido equivalentemente, ou seja, é preciso encontrar s que satisfaça as seguintes equações:

$$\begin{array}{rcl} s.a_1 & \approx_{\mathscr{D}} & b_1 \pmod{R} \\ s.a_2 & \approx_{\mathscr{D}} & b_2 \pmod{R} \\ & \vdots & \end{array}$$

onde a_i e b_i são elementos de R e a redução modular em R é o mesmo que reduzir o polinômio resultante módulo $f(x)$ e seus coeficientes módulo q.

1.4.2. Criptossistema homomórfico restrito

A seguir é apresentado o criptossistema que será utilizado como base da construção final, sendo usada a notação $\mathscr{E}_R(\lambda, \mu)$ para referência a este esquema.

Definição 1.4.2. Configuração. Dado o parâmetro de segurança λ e um parâmetro secundário μ, escolhemos um inteiro q com μ bits e $N = \lceil 3\log q \rceil$.

Geração de chaves. Utilize a distribuição $\mathscr{D}$ para obter o polinômio s^*, denotando por s o vetor de tamanho 2 formado pelos polinômios 1 e s^*. A chave privada é dada por $\mathrm{sk} = s$. Gere aleatoriamente uma matriz A' de N linhas e uma coluna, cujos elementos sejam polinômios com coeficientes uniformemente escolhidos em $\mathbb{Z}_q$. Utilize a distribuição $\mathscr{D}$ para gerar N polinômios e_i e compute $b = A's^* + 2e$. Compute a matriz A de duas colunas, sendo a primeira igual a b e a segunda igual a $-A'$. A chave pública é dada por $\mathrm{pk} = A$. Por construção, temos que $As = 2e$.

Encriptação. Dada uma mensagem $m \in \{0,1\}$, define-se a matriz m' de duas linhas, onde a primeira é o próprio m e a segunda é igual a zero. Gere aleatoriamente a matriz de polinômios binários r, com N linhas. Compute

$$c = m' + A^T r.$$

Decriptação. Compute $m = [[\langle c, s\rangle]_q]_2$.

A corretude deste criptossistema é facilmente verificada usando a relação $As = 2e$ e o fato de q ter sido escolhido suficientemente grande para que o acúmulo de erro não ultrapasse $q/2$, semelhantemente ao caso sobre números inteiros.

1.4.3. Redução de dimensão

O algoritmo de decriptação descrito anteriormente assemelha-se ao ElGamal, porque o texto cifrado é composto por dois polinômios, $c = [c_0, c_1]$, enquanto a chave privada é dada por $s = [1, s^*]$. Portanto, a decriptação pode ser representada por

$$m = [c_0 + c_1 s^*]_q \pmod 2.$$

Interpretando s^* simbolicamente, a expressão $c_0 + c_1 s^*$ representa um polinômio de grau 1. Para multiplicar dois textos cifrados, $c = \mathrm{Enc}(\mathrm{pk}, m)$ e $c' = \mathrm{Enc}(\mathrm{pk}, m')$, podemos computar

$$(c_0 + c_1 s^*)(c'_0 + c'_1 s^*) = c_0 c'_0 + (c_0 c'_1 + c'_0 c_1)s^* + c_1 c'_1 (s^*)^2.$$

Se q for suficientemente grande, ao substituir s pela chave privada na expressão anterior, obtemos um polinômio que pode ser usado para recuperar $m.m'$. Porém, a multiplicação faz com que o texto cifrado esteja em um espaço de dimensão maior. Para que o criptossistema seja compacto, o texto cifrado não pode crescer desta maneira, de modo que é preciso um algoritmo para redução da dimensão. Esta tarefa será realizada por meio de um algoritmo denominado SwitchKey, que, com base em parâmetros públicos, retorna um texto cifrado que pode ser normalmente decriptado.

Dado um polinômio x, considere o algoritmo BitDecomp, que retorna $\log q$ polinômios binários x_i, computados pela representação dos coeficientes de x na base 2. Isto é,

$$x = \sum 2^i x_i.$$

Além disso, considere o algoritmo PowerOf2, que retorna $\log q$ polinômios na forma $2^i x$, como segue:

$$\text{PowerOf2}(x) = [x, 2x, ..., 2^{\lfloor \log q \rfloor} x].$$

Por construção, temos que

$$\langle \text{BitDecomp}(c), \text{PowerOf2}(s) \rangle = \langle c, s \rangle \pmod{q}.$$

Assim, é possível definir da seguinte forma o algoritmo SwitchKeyGen:

1. dado um vetor de polinômios $\bar{s}$, derivado da chave privada s, compute uma nova chave pública $\bar{A}$, correspondente a $\bar{s}$, com $\bar{N}$ linhas, onde $\bar{N} = 3\log^2 q$;
2. retorne $\bar{B} = \bar{A} + \text{PowerOf2}(\bar{s})$, onde $\text{PowerOf2}(\bar{s})$ é adicionado a primeira coluna de $\bar{B}$.

Com isso, dado um texto cifrado expandido $\bar{c}$, o algoritmo SwitchKey pode ser definido simplesmente como

$$\text{SwitchKey}(\bar{c}) = \text{BitDecomp}(\bar{c})^T \bar{B}.$$

Em resumo, a matriz $\bar{B}$ funciona como alternativa ao uso do problema SSP, descrito nos esquemas anteriores. Ou seja, é a encriptação da chave privada usando sua própria chave pública, de modo que estamos novamente assumindo segurança circular. É possível redefinir os algoritmos SwitchKeyGen e SwitchKey, de maneira a utilizar uma cadeia de chaves, mas para simplificar a exposição, foi adotada esta estratégia.

1.4.4. Redução de módulo

Os criptossistemas definidos até agora possuem um problema em comum: o ruído cresce de forma quadrática a cada multiplicação. Para que um esquema seja considerado completamente homomórfico, é necessário que ele seja capaz de avaliar uma quantidade arbitrária de operações de soma ou multiplicação. Portanto, o crescimento quadrático do ruído é um problema que deve ter atenção especial. Para superar este obstáculo, Brakerski e Vaikuntanathan [BV11] propuseram uma nova técnica para gerenciamento do ruído.

Basicamente, se o ruído inicial é proporcional a r, após k multiplicações este ruído passa a ser proporcional a r^{2^k}. A solução encontrada foi utilizar uma

cadeia decrescente de módulos $q_i \approx q/r^i$. Após a primeira multiplicação, ajusta-se o texto cifrado c, multiplicando-o por $1/r$ e corrigindo a paridade se necessário, e troca-se o módulo q por q/r. Esta mudança parece não trazer nenhum ganho e não pode ser realizada arbitrariamente, pois a cadeia decrescente chega rapidamente (linearmente em relação a profundidade do circuito) em um valor mínimo. Porém, é fácil mostrar que o ruído é reduzido na mesma proporção $1/r$, ou seja, após a k-ésima multiplicação, obtemos ruído proporcional a r^k, ao invés de r^{2^k}. Logo, há um ganho exponencial nesta tranformação. Quando a cadeia decrescente chega ao fim, é necessário usar a autoinicialização para retornar ao topo da cadeia.

Sendo assim, dado um vetor de polinômios x, o algoritmo $\text{Scale}(x, q_i, q_{i+1})$ computa o vetor de polinômios mais próximo a $(q_{i+1}/q_i)x$, tal que

$$\text{Scale}(x, q_i, q_{i+1}) = x \pmod 2.$$

1.4.5. BGV

Com isso, definimos nesta seção o esquema ***BGV*** (Brakerski, Gentry, Vaikuntanathan [BGV11]), capaz de avaliar circuitos de profundidade multiplicativa L.

Definição 1.4.3. Configuração. Dado o parâmetro de segurança λ e a profundidade multiplicativa L, compute $\mu = \theta(\log\lambda + \log L)$. Para i variando de L a 0, configure o esquema $\mathscr{E}_{R,i}(\lambda, (i+1)\mu)$, obtendo uma cadeia decrescente de módulos, começando com q_L, que possui $(L+1)\mu$ bits, até q_0, que possui μ bits.

Geração de chaves. Utilize a geração de chaves do esquema $\mathscr{E}_R$ para cada nível i do circuito. Compute $s'_i = s_i \otimes s_i$ e $s''_i = \text{BitDecomp}(s_i, q_i)$, onde BitDecomp recebe q_i por parâmetro, já que agora existem $L+1$ possibilidades para q_i. Finalmente, compute $\bar{B}_i = \text{SwitchKeyGen}(s''_i, s_{i-1})$, para $i > 0$. A chave privada é formada pelos valores de s_i, enquanto a chave pública corresponde às chave públicas de $\mathscr{E}_{R,i}$, acrescidas de $\bar{B}_i$.

Encriptação. Dado o bit m, compute $\mathscr{E}_{R,L}.\text{Enc}(\text{pk}_L, m)$.

Decriptação. Dado um texto cifrado c, no nível k do circuito, utilize a chave privada s_k para computar $m = \mathscr{E}_{R,k}.\text{Dec}(s_k, c)$.

A soma de textos cifrados é realizada pela soma individual dos polinômios, enquanto a multiplicação é realizada pelo produto tensorial dos textos cifrados, obtendo assim um vetor composto por 3 polinômios, denominado texto cifrado expandido, sendo então necessário utilizar o algoritmo Recrypt, definido a seguir, de modo que o texto cifrado volte a ser composto por 2 polinômios.

Dado o texto cifrado expandido $\bar{c}$, q_i e q_{i+1}, o algoritmo Recrypt calcula

$$c_1 = \text{PowerOf2}(\bar{c}, q_i).$$

Neste momento, a seguinte condição é válida: $\langle c_1, s''_j \rangle = \langle c, s'_j \rangle$. Agora, é possível utilizar os algoritmos de redução de dimensão e redução de módulo, isto é, calcula-se $c_2 = \text{Scale}(c_1, q_{i+1}, q_i)$ e a saída do algoritmo é dada por

$$\text{SwitchKey}(c_2, q_i, \bar{B}_i).$$

1.4.6. Operações em bloco

Nesta seção será descrita uma importante otimização sobre o esquema anterior. A ideia consiste em utilizar o teorema chinês dos restos para permitir a operação simultânea sobre um vetor de mensagens. Na literatura, é feita uma analogia ao modelo SIMD, pela capacidade de operar sobre vetores de palavras [SV11].

Com esta otimização é possível reduzir a computação homomórfica de cada nível do circuito para complexidade polilogarítmica, representando assim um grande ganho em relação ao limite anterior de $\Omega(\lambda^{3.5})$. Porém, o circuito a ser avaliado homomorficamente deve ter largura média de $\Omega(\lambda)$.

Além disso, a capacidade de realizar somas e multiplicações sobre vetores não é um modelo computacional completo, porque não é equivalente ao modelo de circuitos algébricos. É necessária uma maneira de permutar os elementos dentro de um determinado vetor, caso contrário não seria possível computar funções relacionando elementos de diferentes posições do vetor. Para resolver este problema foi utilizado o automorfimo de Frobenius, que permite rotacionar os elementos do vetor, e uma rede de permutação, que permite combinar rotações a esquerda e a direita para realizar permutações mais complicadas.

Matematicamente, sejam m e q inteiros tais que $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ contém uma raíz m-ésima primitiva da unidade, $w \in \mathbb{Z}_q$, então o m-ésimo polinômio ciclotômico, $\Phi_m(x)$, pode ser fatorado em termos lineares módulo q

$$\Phi_m(x) = \prod (x - w^i) \pmod{q}.$$

Um polinômio em $\mathbb{Z}_q[x]/\Phi_m(x)$ pode ser representado por seus coeficientes, como vínhamos fazendo até agora, ou então pode ser representado por sua avaliação exatamente nas raízes m-ésimas primitivas da unidade. Assim, existem duas possíveis representações: por coeficientes, ou por avaliação. A primeira denotaremos por $\overrightarrow{c}$, enquanto a segunda denotaremos por $\overrightarrow{d}$, de modo que as representações estão relacionadas pela matriz de Vandermonde V_m da seguinte maneira

$$\overrightarrow{d} = V_m \overrightarrow{c}.$$

Considerando q como o produto de primos p_i, podemos de fato utilizar duas vezes o teorema chinês dos restos (TCR), já que os próprios elementos de $\mathbb{Z}_q$ podem ser decompostos de acordo os fatores de q. Esta representação é chamada de **TCR dupla**.

1.4.7. AES homomórfico

Recentemente [GHS12b], Craig Gentry Shai Halevi e Nigel Smart apresentaram um resultado prático importante: a avaliação homomórfica do AES-128. A implementação foi realizada utilizando a biblioteca NTL sobre GMP. Foi usado o esquema BGV, com a representação TCR dupla, de forma que o modelo SIMD foi utilizado para computar sobre os blocos do AES. A execução de uma rodada completa do AES demorou aproximadamente uma semana, realizada por um computador com 256 GB de memória RAM.

A escolha do AES é de especial importância, porque por um lado possui estrutura que permite o uso das técnicas descritas na seção anterior, para operações em bloco, e, por outro lado, permite transformar um texto cifrado obtido pelo AES em um texto cifrado homomórfico, de modo que o dado pode ser manipulado sem que em nenhum momento o fique desprotegido, isto é, em claro.

Este trabalho utiliza a versão mais eficiente de ECH construída até o momento, propondo uma técnica nova para o gerenciamento dinâmico do ruído. Uma estimativa do ruído é acrescentada ao texto cifrado. Assim, foi possível minimizar a quantidade de vezes que a reencriptação é necessária. Por sua vez, isto é interessante porque este algoritmo precisa transformar da representação TCR dupla para a representação em coeficientes e esta transformação requer bastante processamento. De fato, a transformação entre representações é realizada pelo algoritmo FFT e FFT inverso de um vetor de polinômios. Com isso, este trabalho pode ser considerado como um marco importante para a encriptação homomórfica.

1.5. Aplicações

Existem diversas aplicações, tanto práticas como teóricas para homomorfismos secretos. Muitas dessas aplicações não requerem a existência de homomorfismos completos, isto é, que permitem uma quantidade arbitrária de operações de soma e multiplicação. Na tese de doutorado de Dörte K. Rappe [Rap06], são descritas diversas aplicações de criptossistemas baseados em homomorfismos. Fazemos aqui um breve resumo dessas aplicações.

1.5.1. Agentes móveis

Uma aplicação interessante do uso de homomorfismo secreto é na proteção de agente móveis [SST97]. Uma preocupação neste cenário é a possibilidade de ataques de um servidor malicioso para obtenção de dados sigilosos ou para deduzir informações a respeito de determinada computação. Com o uso de criptografia baseada em homomorfismos, é possível computar sobre dados criptografados ou então é possível computar sobre funções criptografadas. Dependendo do cenário em questão, é possível utilizar a alternativa mais adequada.

1.5.2. Computação multiparte

Neste cenário, um grupo de indíviduos estão interessados em calcular uma função f, de modo que cada indivíduo contribua com uma parte dos parâmetros de

entrada da função, e tal que terceiros, que previamente não tinham conhecimento desses parâmetros, não passem a conhecê-los após a computação de f. Homomorfismos secretos permitem que a função f seja computada utilizando a forma encriptada dos parâmetros de entrada, obtendo como retorno a encriptação da saída de f computada sobre os parâmetros em claro.

1.5.3. Compartilhamento de segredo

Neste contexto, o homomorfismo algébrico implica que a composição dos segredos compartilhados é igual ao compartilhamento dos segredos compostos, solucionando o problema de forma elegante.

1.5.4. Assinaturas

Dado um conjunto de ***assinaturas*** válidas para um determinado conjunto de dados, é possível construir uma nova assinatura, que correspondente a avaliação de uma função f, sobre um subconjunto desses dados. Este é um tema que tem sido pouco explorado, como argumenta Patrick Schmidt [Sch11] em sua dissertação. Em outro trabalho interessante [BF11], Dan Boneh e David Freeman apresentam um esquema capaz de avaliar uma classe restrita de funções. A proposta é semelhante a ECH sobre reticulados ideais de Craig Gentry.

1.5.5. Conhecimento nulo

De forma simplificada, em contextos que utilizam o conceito de *conhecimento nulo* alguém (Bob) deseja demonstrar (para Alice) que possui uma determinada informação sem que seja necessário revelá-la. De fato, Bob deseja que **nenhuma** informação seja revelada. Esta primitiva criptográfica tem grande importância teórica e prática. A utilização de homomorfismos permite que Bob encripte a informação de forma que Alice ainda consiga validar uma determinada propriedade algébrica desta informação. Como o homomorfismo preserva a estrutura algébrica, a encriptação da informação preserva as suas propriedades algébricas.

1.5.6. Eleições

Este é um contexto de peculiar importância já que, cada vez mais, países estão utilizando urnas eletrônicas na escolha de seus governantes. Este é um exemplo de cenário em que não é necessária a utilização de um homomorfismo completo, já que deseja-se apenas somar 1 a uma quantia de votos, mas não é necessário multiplicar quantias de votos. Sendo assim, esquemas parcialmente homomórficos são suficientes para tornar esta aplicação prática. Nos sistemas de votação Votebox e Helios [SDW], é utilizada uma adaptação do criptossistema ElGamal para permitir contagem dos votos encriptados, de modo a garantir o sigilo de cada voto e decriptar o resultado apenas na hora da contagem de votos, utilizando a chave secreta do esquema.

1.5.7. Ofuscação

Em um outro trablho [DMMQN11] é apresentada uma proposta para o uso de ECH no contexto de ofuscação. Contudo, usando ECH o programa ofuscado

produz como saída um texto cifrado. Para lidar com a situação, o receptor do programa ofuscado precisa ser capaz de provar que não é malicioso, para, com o auxílio de um hardware com características especiais, conseguir decifrar a saída do programa.

Propostas anteriores não permitiam múltiplas execuções do mesmo programa ofuscado, ou então precisam de um hardware distinto para cada nível do circuito a ser avaliado. Portanto, mesmo com os resultados negativos sobre a possibilidade real de ofuscação, a ECH permite a construção de esquemas melhores quando se supõe um modelo de segurança menos restritivo.

1.5.8. Encriptação parcialmente homomórfica

As otimizações propostas, principalmente sobre o esquema BGV, além de contribuir diretamente na tarefa de tornar prática a ECH, permitem a construção de esquemas de encriptação parcialmente homomórfica (EPH), capazes de solucionar diversos problemas práticos, como é mostrado em [NLV11b]. Neste trabalho, uma prova de conceito é desenvolvida na linguagem aritmética Magma, mostrando que existe uma liberdade na escolha de parâmetros do esquema BGV, de modo que é possível adaptá-lo de acordo com o circuito a ser avaliado homomorficamente. Diversas escolhas de parâmetros são sugeridas para diferentes cenários, dependendo da possibilidade de uso de operações em bloco, da quantidade de multiplicações envolvidas e da profundidade do circuito.

Recentemente [PRZB11], é apresentado o sistema CryptDB, que é um banco de dados sobre dados cifrados. São utilizadas diversas primitivas criptográficas para permitir consultas SQL arbitrárias. Existem algumas limitações em relação a certos tipos de junções, que na prática são pouco frequentes. As operações foram muito bem definidas e organizadas nos seguintes grupos: (i) verificação de igualdade; (ii) comparação de ordem; (iii) operações aritméticas; e (iv) junções. O esquema é formado por camadas aninhadas de cifras para resolver cada um desses grupos. Para a execução de operações aritméticas é usado o criptosisstema homomórfico Paillier, que é capaz de efetuar somas (mas não permite multiplicações). A construção oferece confidencialidade, considerando um modelo de adversário passivo, mas os próprios autores supõem uma segurança que não é perfeita, porque o adversário consegue, por exemplo, ordenar os dados. Por outro lado, é uma abordagem prática interessante, porque além de eficiente também é transparente para o usuário, já que o servidor interpreta dinamicamente as consultas SQL, mapeando-as em funções internas do banco de dados. Com isso, é possível oferecer proteção contra o próprio administrador do banco de dados.

Outro trabalho relacionado é a proposta de uma linguagem de domínio específico (*domain specific language*) para computação em nuvem [BMS+11]. É utilizada a linguagem funcional Haskell em conjunto com um esquema de encriptação parcialmente homomórfica para construção de uma plataforma para execução segura de código, permitindo oferecer confidencialidade das informações. Este trabalho é semelhante ao CryptDB, na medida em que se supõe um modelo de segurança menos rígido para resolver um problema com um escopo

bem determinado.

1.6. Considerações finais

Neste minicurso foram apresentados os recentes trabalhos de Craig Gentry, que resolveram um problema que permaneceu em aberto por 31 anos, que principalmente hoje em dia, com a consolidação do modelo de computação em nuvem, oferece uma solução elegante ao permitir a computação sobre dados encriptados.

A construção representa um avanço teórico, pela solução da conjectura feita por Rivest, Adleman e Dertouzos em 1978, reunindo uma variedade de conceitos matemáticos interessantes. É importante ressaltar que os esquemas propostos possuem demonstração de segurança, com base em problemas difíceis em reticulados, um assunto que ganhou novamente a atenção da comunidade científica por resistir à ataques quânticos, isto é, que fazem uso de computadores quânticos. Além disso, as construções ainda podem ser adaptadas de acordo com o problema a ser resolvido. Sendo assim, o material aqui exposto reuniu o estado da arte em encriptação homomórfica, mostrando uma série de trabalhos recentes, que representam um grande avanço para a criptografia moderna.

Como vimos, apesar de todas as otimizações propostas, a encriptação completamente homomórfica ainda é inviável para ser usada na prática. Existem resultados negativos [Bra12] que confrontam a capacidade homomórfica de um criptossistema com a eficiência do algoritmo de decriptação, estabelecendo assim um limite inferior para a computação da autoinicialização. Concretamente, se o esquema for capaz de avaliar homomorficamente a função de majoridade, então a decriptação não pode ser linear.

Existem alguns problemas em aberto, dentre os quais vale destacar: a construção de um esquema de ECH que não seja baseado na existência de ruído. Em especial, a multiplicação de textos cifrados resulta em um elemento com dimensão maior que os valores iniciais. Encontrar uma forma de multiplicar sem que isto ocorra é um problema interessante em aberto.

Uma outra linha de pesquisa possível é sobre o uso de encriptação parcialmente homomórfica. Foi mostrado que o esquema BGV pode ser adaptado para diferentes profundidades multiplicativas, permitindo encontrar uma boa configuração para diversos problemas práticos. Além disso, também foram apresentados outros criptossistemas que podem ser utilizados em alguns cenários, como por exemplo com o uso do ElGamal no contexto de eleições eletrônicas no projeto VoteBox.

A tabela a seguir mostra a complexidade por operação homomórfica, após otimizações, para esquemas de encriptação homomórfica sobre inteiros e reticulados, além da recente proposta com base no problema RLWE. Diversos trabalhos estão surgindo propondo modificações ao esquema BGV, obtendo vantagem em determinados cenários [GHPS12, GHS12a].

Versão	Complexidade a cada operação homomórfica
Inteiros	$O(\lambda^5)$
Reticulado	$O(\lambda^{3.5})$
RLWE (BGV)	$\tilde{O}(\lambda)$

1.7. Exercícios

1. Considerando o esquema simétrico sobre inteiros, onde a encriptação é calculada por $c = m + 2r + pq$, se $|2r| < 50$, $p = 5001$ e $10001 \leq q \leq 20001$. Na pior das hipóteses, quantas somas podemos realizar homomorficamente? E com relação às multiplicações?

2. Modificando apenas o valor de p no exercício anterior, calculamos os textos cifrados $c = 79818018$ e $c' = 80616104$. Implemente um programa para encontrar o novo valor de p, sabendo que possui a mesma quantidade de dígitos decimais que o valor anterior. Descubra também as mensagens m e m', correspondentes a c e c', respectivamente.

3. O esquema das questões anteriores pode ser facilmente adaptado para permitir espaço de texto claro $\mathbb{F}_3$. Para isso, a encriptação é realizada por $c = m + 3r + pq$, além disso, é necessário que $|3r| < 50$. Se o restante das condições permanecerem iguais, é possível multiplicar homomorficamente?

4. Para que o espaço de texto claro seja $\mathbb{F}_3$ o algoritmo de decriptação computa $m = ((c \pmod p) \pmod 3)$. Altere o programa feito no exercício 2 para usar este espaço de texto claro e compute quais seriam os respectivos valores de m e m'.

5. Considerando $f(x) = x^2 - 1$ e o anel $\mathbb{Z}[x]/f(x)$, responda as seguintes questões:

   (a) Quais são as classes laterais do ideal (2)?

   (b) Compute a base de rotação B do reticulado gerado por $(a(x))$, onde $a(x) = x + 2$.

   (c) O polinômio $p(x) = 15x + 12$ pertence ao reticulado gerado por $(a(x))$?

   (d) Compute $x + 10 \pmod B$.

6. Seja $R = \mathbb{Z}[x]/f(x)$, onde $f(x) = x^2 - 1$. Considerando o ideal polinomial J gerado por $(a(x))$, onde $a(x) = 5001x + 10002$. Dado par de chaves (B_J^{sk}, B_J^{pk}) a seguir

$$\left(\begin{bmatrix} 10002 & 5001 \\ 5001 & 10002 \end{bmatrix}; \begin{bmatrix} 15003 & 5001000 \\ 0 & 10002 \end{bmatrix} \right)$$

   e o texto cifrado $c = [-14980, 37]^T$, responda aos seguintes itens:

(a) Compute c^2, lembrando que c corresponde ao polinômio $c(x) = 37x - 14980$.

(b) Compute $c^2 \pmod{B_J^{pk}} \pmod 2$.

(c) O que é possível concluir a respeito de c?

7. No exercício anterior, a encriptação de uma mensagem m, interpretada como polinômio em $\mathbb{Z}[x]/(x^2-1)$, é calculada somando-se a m um polinômio da forma $2r_1x + 2r_0$, onde $|2r_i| < 50$, para $i \in \{0,1\}$. Quantas multiplicações são permitidas por este esquema?

8. Implemente um programa para computar a chave secreta utilizada no exercício 6.

9. Considerando o esquema BGV sobre o anel $R_q = \mathbb{Z}_q[x]/(x^2+1)$, para $q = 5001$ e $N = 2$. Dada a chave pública

$$A = \begin{bmatrix} -1072x - 1604 & -1367x - 259 \\ -1310x + 326 & -521x + 811 \end{bmatrix},$$

a encriptação é computada por $c = m + A^T r$, onde r é um vetor coluna de polinômios em R_2, isto é, com coeficientes em $\{-1,0,1\}$. Dado o texto cifrado $c = [-2168x + 1693, -224 + 1916]^T$, responda aos seguintes itens:

(a) Compute c^2, lembrando que $c = [c_0, c_1]$ e pode ser interpretado como um polinômio $c(v) = c_0 + c_1 v$, onde v é uma variável simbólica nova. Assim, $c^2 = (c_0 + c_1 v)^2 = c_0^2 + 2c_0c_1v + c_1^2v^2$. Ou seja, para computar c^2, é preciso computar os coeficientes c_0^2, $2c_0c_1$ e c_1^2.

(b) Compute $\langle c^2, s \otimes s \rangle_q \pmod 2$.

(c) O que é possível concluir a respeito de c?

10. Sabendo que a chave pública do exercício anterior foi gerada usando erro e tal que $|2e| < 50$, quantas multiplicações são permitidas?

11. Implemente um programa para computar a chave secreta utilizada no exercício 9.

1.8. Material complementar

Para complementar este minicurso, foram reunidos exemplos, exercícios, textos e referências sobre encriptação homomórfica. Este material encontra-se disponível na web, no endereço `http://www.fhe.tecic.com.br`.

Referências

[Bab86] L Babai. On lovász lattice reduction and the nearest lattice point problem. *Combinatorica*, (6), 1986.

[BF11] Dan Boneh and David Freeman. Homomorphic signatures for polynomial functions, 2011.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, 2001.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In Joe Killian, editor, *Proceedings of Theory of Cryptography Conference 2005*, volume 3378 of *LNCS*, pages 325–342. Springer, 2005.

[BGV11] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. Fully homomorphic encryption without bootstrapping. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:111, 2011.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50:506–519, July 2003.

[BMS+11] Alex Bain, John Mitchell, Rahul Sharma, Deian Stefan, and Joe Zimmerman. A Domain-Specific Language for Computing on Encrypted Data (Invited Talk). In Supratik Chakraborty and Amit Kumar, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011)*, volume 13 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6–24, Dagstuhl, Germany, 2011. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Bra12] Zvika Brakerski. When homomorphism becomes a liability. Cryptology ePrint Archive, Report 2012/225, 2012. `http://eprint.iacr.org/`.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. Cryptology ePrint Archive, Report 2011/344, 2011. `http://eprint.iacr.org/`.

[CMNT11] Jean-Sébastien Coron, Avradip Mandal, David Naccache, and Mehdi Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In *Proceedings of the 31st annual conference on Advances in cryptology*, CRYPTO'11, pages 487–504, Berlin, Heidelberg, 2011. Springer-Verlag.

[CNT11] Jean-Sebastien Coron, David Naccache, and Mehdi Tibouchi. Optimization of fully homomorphic encryption. Cryptology ePrint Archive, Report 2011/440, 2011. http://eprint.iacr.org/.

[DF04] D.S. Dummit and R.M. Foote. *Abstract algebra*. Wiley, 2004.

[DH76] W Diffie and M E Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, (22), 1976.

[DMMQN11] Nico Döttling, Thilo Mie, Jörn Müller-Quade, and Tobias Nilges. Basing obfuscation on simple tamper-proof hardware assumptions. *IACR Cryptology ePrint Archive*, 2011:675, 2011.

[FK94] M. Fellows and N. Koblitz. Combinatorial cryptosystems galore! In G. L. Mullen and P. J.-S. Shiue, editors, *Finite Fields: Theory, Applications, and Algorithms*, volume 168 of *Contemporary Mathematics*, pages 51–61. 1994.

[Gen09a] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. crypto.stanford.edu/craig.

[Gen09b] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 169–178, New York, NY, USA, 2009. ACM.

[GGH] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *In CRYPTO, pages 112–131, 1997.*

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *In CRYPTO*, pages 112–131, 1997.

[GH11a] Craig Gentry and Shai Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In *FOCS*, pages 107–109, 2011.

[GH11b] Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme. In Kenneth Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 129–148. Springer Berlin / Heidelberg, 2011. 10.1007/978-3-642-20465-4_9.

[GHPS12] Craig Gentry, Shai Halevi, Chris Peikert, and Nigel P. Smart. Ring switching in bgv-style homomorphic encryption. Cryptology ePrint Archive, Report 2012/240, 2012. http://eprint.iacr.org/.

[GHS12a] Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully homomorphic encryption with polylog overhead. In *EUROCRYPT*, pages 465–482, 2012.

[GHS12b] Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the aes circuit. *IACR Cryptology ePrint Archive*, 2012:99, 2012.

[GM82] S. Goldwasser and S. Micali. Probabilistic Encryption and How To Play Mental Poker Keeping Secret All Partial Information. In *Proc. 14th ACM Symp. on Theory of Computing*, pages 270–299. ACM, 1982.

[HG01] Nick Howgrave-Graham. Approximate integer common divisors. In *CaLC*, pages 51–66, 2001.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *Advances in Cryptology EUROCRYPT 2010*, 6110/2010(015848):1?23, 2010.

[MG09] Peter Mell and Tim Grance. The nist definition of cloud computing. *National Institute of Standards and Technology*, 53(6):50, 2009.

[NLV11a] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, CCSW '11, pages 113–124, New York, NY, USA, 2011. ACM.

[NLV11b] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, CCSW '11, pages 113–124, New York, NY, USA, 2011. ACM.

[PRZB11] Raluca Ada Popa, Catherine M. S. Redfield, Nickolai Zeldovich, and Hari Balakrishnan. Cryptdb: protecting confidentiality with encrypted query processing. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, pages 85–100, New York, NY, USA, 2011. ACM.

[RAD78] R L Rivest, L Adleman, and M L Dertouzos. On data banks and privacy homomorphisms, in r. a. demillo et al. In *Eds.), Foundations of Secure Computation*, pages 169–179. Academic Press, 1978.

[Rap06] Doerte K. Rappe. Homomorphic cryptosystems and their applications. Cryptology ePrint Archive, Report 2006/001, 2006. `http://eprint.iacr.org/`.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.

[RSA83] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 26:96–99, January 1983.

[Sch11] Patrick Schmidt. Fully homomorphic encryption: Overview and cryptanalysis. Diploma thesis, TU Darmstadt, Jul 2011.

[SDW] Daniel Sandler, Kyle Derr, and Dan S. Wallach. Votebox: a tamper-evident, verifiable electronic voting system.

[SS10] Damien Stehle and Ron Steinfeld. Faster fully homomorphic encryption. Cryptology ePrint Archive, Report 2010/299, 2010. `http://eprint.iacr.org/`.

[SST97] Tomas Sander, Tomas S, and Christian F. Tschudin. Protecting mobile agents against malicious hosts, 1997.

[SV09] N.P. Smart and F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. Cryptology ePrint Archive, Report 2009/571, 2009. `http://eprint.iacr.org/`.

[SV11] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic simd operations. *IACR Cryptology ePrint Archive*, 2011:133, 2011.

[vDGHV09] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. Cryptology ePrint Archive, Report 2009/616, 2009. `http://eprint.iacr.org/`.

Índice Remissivo

2.2 Key recovery attacks

In this section we propose key-recovery attacks to the NTRU-based family of SHE schemes. This work closes a gap in the literature, because this family was the last one to be shown insecure against CCA1 adversaries. The paper remarks that decryption oracle queries are a real threat in the cloud computing scenario, and, since only a few queries are necessary to totally break the scheme by computing the private key, it is crucial to deal with this attack in order to obtain homomorphic encryption schemes that are appropriate for outsourcing computation to the cloud. This is done in the next section.

This paper was published at the International Conference on Information-Theoretic Security, Lugano, Switzerland, in 2015.

Adaptive Key Recovery Attacks on NTRU-based Somewhat Homomorphic Encryption Schemes

Ricardo Dahab[1] *, Steven Galbraith[2], and Eduardo Morais[1] **

[1] Institute of Computing, University of Campinas, Brazil

[2] Mathematics Department, University of Auckland, New Zealand

Abstract. In this paper we present adaptive key recovery attacks on NTRU-based somewhat homomorphic encryption schemes. Among such schemes, we study the proposal by Bos et al [BLLN13] in 2013. Given access to a decryption oracle, the attack allows us to compute the private key for all parameter choices. Such attacks show that one must be very careful about the use of homomorphic encryption in practice. The existence of a key recovery attack means that the scheme is not CCA1-secure. Indeed, almost every somewhat homomorphic construction proposed till now in the literature is vulnerable to an attack of this type. Hence our result adds to a body of literature that shows that building CCA1-secure homomorphic schemes is not trivial.

1 Introduction

The construction of *fully homomorphic encryption* (FHE) was conjectured in 1978 by Rivest, Adleman and Dertouzos [RAD78]. Although it was immediately recognized as a very interesting possibility in cryptography, no concrete construction was known until 2009, when Gentry used ideal lattices to settle this conjecture [Gen09a].

In short, ciphertexts produced by an FHE scheme can be operated on in such a way that we obtain a ciphertext that corresponds to the addition or multiplication of the respective plaintexts. The ability to algebraically operate over ciphertexts is of great importance because we can transform any algorithm into a sequence of additions and multiplications in $\mathbb{Z}_2$. Therefore, such a scheme can evaluate any algorithm solely with access to the encryption of its input, and such that the computation returns the encryption of the output.

Since Gentry's work, many FHE constructions have appeared in the literature. However, all the proposals have a common drawback: they are not practical. Initially, the algorithms involved in the constructions, although having polynomial complexity, had high polynomial degree. Later, the asymptotic complexity became much better. Indeed, we now have constructions with polylog overhead per operation, but with terribly high constants.

* Partially supported by CNPq grant 311530/2011-7, and FAPESP Thematic Project 2013/25977-7

** Partially supported by FAPESP Thematic Project 2013/25977-7

Although fully homomorphic encryption is not practical yet, many constructions have been proposed recently, achieving a somewhat homomorphic encryption (SHE) scheme. They allow a limited "depth" of operations to be performed. These constructions are indeed very useful in practice, specially in order to provide security in the scenario of cloud computing. SHE is important also in the implementation of *private information retrieval* (PIR) protocols, which can be seen as a building block to the solution for the privacy problem that emerges when we give our data to the cloud.

In the cloud computing scenario it is natural to imagine an attacker having access to a decryption oracle (e.g., the cloud can feed invalid ciphertexts to a user and monitor their behaviour). It is obvious that a homomorphic encryption scheme cannot have security of ciphertexts under adaptive attacks. Hence, adaptive attacks are already a very serious concern in this setting. But one could hope that at least the private key remains secure in the presence of a decryption oracle. However, it is already known that this is not necessarily the case. Loftus et al [LMSV12] were the first to observe adaptive key recovery attacks, and further examples were given by Zhang et al [ZPS12] and Chenal and Tang [CT14]. By now, most schemes have been attacked, but the NTRU-based schemes remained unbroken.

Gentry's original construction is based on ideal lattices and is naturally implemented using cyclotomic rings. On the other hand, NTRU is a practical lattice-based cryptosystem, also based on cyclotomic rings, that remained without a security proof for a long time. Recently NTRU was put on a stronger foundation by Stehlé and Steinfeld [SS11], and NTRU-based cryptosystems returned as a fruitful research area. Scale-invariant homomorphic encryption was proposed by Brakerski [Bra12], presenting a construction that avoids the utilization of modulus switching technique, considerably simplifying the scheme.

In this work, we present *adaptive key recovery* attacks on NTRU-based SHE schemes. In particular, we attack the *scale-invariant* proposal by Bos et al [BLLN13].

1.1 Notation

Notation $\lfloor a \rceil$ is used to round a to the nearest integer, while notation $[a]_q$ is used to denote centralized modular reduction, i.e. reduction modulo q, but with result given in the interval $(-q/2, q/2]$. If a is a polynomial, then in order to compute $[a]_q$ we must compute a centralized modular reduction of each coefficient of a (analogously for $\lfloor a \rceil$). When working over a polynomial ring R, if $a(x) \in R$, we use the notation $a[i]$ to denote the i-th coefficient of the polynomial $a(x)$.

1.2 Paper Organization

This paper is organized as follows. In section 2 we present basic definitions and details about the security model that will be used. In section 3 we gather information about key recovery attacks on other schemes in the literature. In section 4 we describe exactly how the SHE scheme BLLN is constructed. In section 5 we provide the main contribution of this paper, which is the key recovery attack. Finally, in section 6 we give our concluding remarks.

2 Fundamentals and Security Model

In this section we are going to present basic concepts and the security model that we will use throughout the paper.

Definition 1. Homomorphic encryption. *A homomorphic cryptosystem is defined using four algorithms,* KeyGen, Dec, Enc, Eval. *The first three are conventional encryption algorithms, with plaintext space $\mathcal{P}$ and security parameter λ. The scheme is said to be* correct *if, for a given algebraic circuit C, every key pair $(\mathrm{sk}, \mathrm{pk})$ generated by* $\textsc{KeyGen}(\lambda)$*, any message tuple $(m_1, \ldots, m_t) \in \mathcal{P}^t$ and corresponding ciphertexts $\Psi = \langle \psi_1, \ldots, \psi_t \rangle$, that is, $\psi_i = \textsc{Enc}_{\mathrm{pk}}(m_i)$ for $1 \leq i \leq t$, then we have that the* Eval *algorithm respects the following relation*

$$\textsc{Dec}_{\mathrm{sk}}(\textsc{Eval}_{\mathrm{pk}}(C, \Psi)) = C(m_1, \ldots, m_t).$$

Furthermore, the algorithms KeyGen, Dec, Enc *and* Eval *must have polynomial complexity and we say that the scheme is homomorphic with respect to the circuit C.*

Definition 2. Fully Homomorphic Encryption. *A scheme $\mathcal{E} = (\textsc{KeyGen}, \textsc{Dec}, \textsc{Enc}, \textsc{Eval})$ is* correct for a class $\mathbf{S_C}$ *of circuits, if it is correct for each $\mathbf{C} \in \mathbf{S_C}$. Moreover, $\mathcal{E}$ is called* fully homomorphic encryption *(FHE) scheme, if it is correct for every algebraic circuit. Alternatively, we can base our construction over Boolean circuits, because both computational models are equivalent. If the scheme can deal with a restricted class of circuits, but not every one, then we call the scheme a* somewhat homomorphic encryption *(SHE) scheme.*

A cryptosystem is secure against *chosen ciphertext attack* (CCA2) if there is no polynomial time adversary $\mathcal{A}$ that can win the following game with non negligible probability.

Setup. The challenger obtains $(\mathrm{sk}, \mathrm{pk}) = \textsc{KeyGen}(\lambda)$ and sends pk to adversary $\mathcal{A}$.

Queries. $\mathcal{A}$ sends ciphertexts to the challenger, before or after the challenge. The challenger returns the corresponding plaintexts.

Challenge. The adversary randomly generates two plaintexts $m_0, m_1 \in \mathcal{P}$ and sends them to the challenger, who chooses randomly a bit $b \in \{0, 1\}$ and computes the ciphertext $c = \textsc{Enc}_{\mathrm{pk}}(m_b)$. The challenger sends c to $\mathcal{A}$.

Answer. $\mathcal{A}$ sends a bit b' to the challenger and wins the game if $b' = b$.

If we allow queries only before the challenge, we say that the cryptosystem is secure against CCA1 adversaries (lunchtime attacks). As previously described, queries can be interpreted as access to a decryption oracle. If instead we only allow access to an encryption oracle, i.e., the adversary can choose any message that is distinct from m_0 and m_1 to be encrypted under the same key pair, then we say that the cryptosystem is secure against *chosen plaintext attacks* (CPA).

In homomorphic encryption, it is impossible to achieve CCA2 security, because the adversary can add an encryption of zero to the encrypted challenge, or multiply it by the encryption of one, and send it to the decryption oracle, which allows him to trivially win the game. Many FHE schemes have as public value an encryption of the private key bits, which can be sent to the decryption oracle before the challenge, which makes such schemes insecure against CCA1 adversaries. Indeed, a *key recovery* attack is stronger than a CCA1 attack and Loftus et al [LMSV12] showed that Gentry's construction over ideal lattices is vulnerable to it and presented the only SHE proposal that is known to be CCA1 secure.

Recently [CT14], Chenal and Tang showed that many SHE schemes are not CCA1 by presenting a key recovery attack. The aim of this paper is to consider such attacks in the setting of NTRU-based schemes.

From now on we are going to work over the cyclotomic ring $R_q = \mathbb{Z}_q[x]/(x^d + 1)$, where d is a power of 2. Cyclotomic rings were introduced to lattice-based cryptography in [HPS98], and have been very popular since the breakthrough work of Lyubashevsky et al [LPR13]. Lattices constructed using such rings are often called *ideal lattices*. Although there is no proof that ideal lattices maintain the same security guarantees as conventional lattices, no significant improvement in the complexity of algorithms for computational problems in ideal lattices is known.

3 Previous Constructions

We can divide homomorphic encryption schemes as in Figure 1. In the first column, we have the schemes that are based on integers, which are simpler to understand. Lattice-based con-

structions are separated in four categories: the initial schemes, that still depend on the Sparse Subset Sum Problem (SSSP); Brakerski-Gentry-Vaikuntanathan (BGV)-like proposals, that bring new concepts and allow better constructions in practice; asymptotically better constructions that are based on the *approximate eigenvector* method, and NTRU-based schemes, that permit to obtain ciphertexts that correspond to just one ring element, simplifying previous schemes. NTRU-based SHE offers the possibility of encoding integers in a natural way, that can be used to solve practical problems such as statistical applications [LLAN14, BLN14].

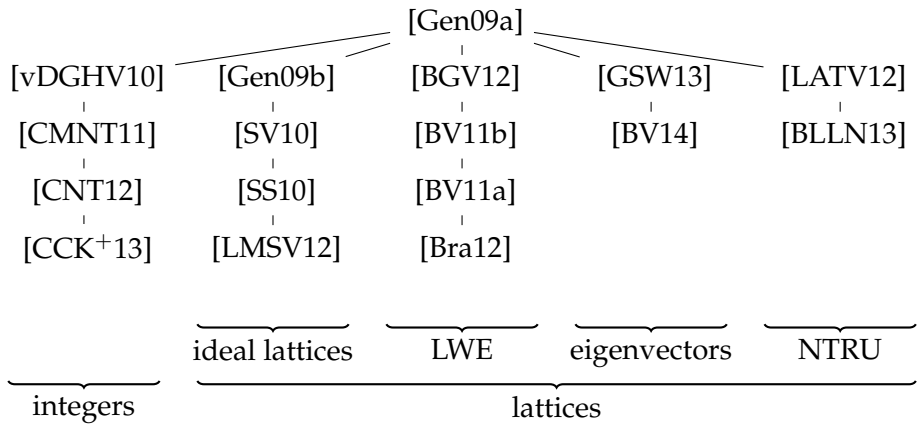

Fig. 1. Homomorphic Encryption Proposals

In the literature [ZPS12, LMSV12, CT14] there are adaptive key recovery attacks on many schemes and these schemes were adapted and optimized later; thus, such constructions should be assessed in order to verify whether the attacks are still feasible. Table 1 shows which schemes have been attacked by each of the previously cited works, showing also which schemes seem to be vulnerable to the same kind of attacks. Although some of them were not directly attacked, the key generation and decryption algorithms are so close to the attacked schemes, that the same strategy can be followed to compute the private key using decryption oracles.

Attack	Schemes	Seems to extend to
[ZPS12]	[vDGHV10, CMNT11]	[CNT12]
[LMSV12]	[Gen09b, SV10, GS11]	[SS10]
[CT14]	[vDGHV10, BGV12, BV11b, BV11a, Bra12, GSW13]	[BV14]
this work	[LATV12, BLLN13]	-
no attack	[LMSV12]	-

Table 1. Key recovery attacks

4 NTRU-based Somewhat Homomorphic Encryption

NTRU [HPS98] is an efficient lattice-based cryptographic scheme but, for many years, the lack of security proofs, reducing its security to worst-case hard lattice problems, was a serious concern. Stehlé and Steinfeld [SS11] presented such a proof, replacing the original ring $\mathbb{Z}_q[x]/(x^d - 1)$ by the previously described cyclotomic ring $R_q = \mathbb{Z}_q[x]/(x^d + 1)$, where d is restricted to a power of 2.

In 2012, López-Alt, Tromer and Vaikuntanathan [LATV12] proposed the construction of *multikey fully homomorphic encryption*, which we call the LTV scheme. The difference here is that users with distinct keys can compute ciphertexts that will be processed by a server in order to obtain the homomorphic evaluation of a determined function. It means that all the users together will be able to decrypt the function evaluation and this strategy can be followed to construct a multiparty computation scheme. Doröz, Hu and Sunar [DHS14] implemented the LTV scheme. They implemented also the homomorphic evaluation of AES, showing that it offers advantages against the BGV scheme [BGV12].

However, the LTV scheme is based on non-standard assumptions. In 2013, a scale-invariant NTRU-based scheme was proposed by Bos et al [BLLN13]. We call it the BLLN scheme. The basic scheme, $\mathcal{E}_{\text{basic}}$, can be described as follows:

Definition 3. ***Setup.*** *Given the security parameter λ construct the ring $R = \mathbb{Z}[x]/(x^d + 1)$, where d is a power of two. Define $R_q = R/(q) \cong \mathbb{Z}_q[x]/(x^d + 1)$. Choose a small integer t, real numbers σ_k and σ_e and a prime q such that $t, \sigma \ll q$. Let $\mathcal{D}_{\text{key}}$ and $\mathcal{D}_{\text{err}}$ be distributions on R coming from discrete Gaussians on $\mathbb{Z}$ with standard deviations σ_k and σ_e respectively. The* SETUP *algorithm returns $(t, d, q, \mathcal{D}_{\text{key}}, \mathcal{D}_{\text{err}})$.*

Key generation. *Given the output of the* SETUP *algorithm, sample polynomials $f', g \leftarrow \mathcal{D}_{\text{key}}$ and compute $f = [tf' + 1]_q$. Check that f is invertible modulo q, if not choose a new f'. Compute the inverse $f^{-1} \in R_q$ and set $h = [tgf^{-1}]_q$. The public key is $pk = h$ and the private key is $sk = f$. Algorithm* KEYGEN *returns (sk, pk).*

Encryption. *The plaintext space is R/tR, so a message is given by a coset $m + tR$. Let $[m]_t$ be a canonical representative element of the coset. Sample $s, e \leftarrow \mathcal{D}_{\text{err}}$ and compute the ciphertext*

$$c = \text{ENC}_{pk}(m) = \left[\lfloor q/t \rfloor [m]_t + te + hs\right]_q .$$

Decryption. *Compute*

$$m = \text{DEC}_{sk}(c) = \left[\lfloor (t/q).[fc]_q \rceil\right]_t .$$

Return the message $[m]_t$.

Given the integers t and q returned by the SETUP algorithm, the plaintext space is given by R/tR, while the ciphertext space is given by R/qR. Note that $t \ll q$. Indeed, the last condition is important to enable as many multiplications as possible. Thus, if t grows when compared to a fixed q, then we would be able to execute fewer multiplications. Although the multiplicative depth of a homomorphic encryption scheme is an important issue, it is not relevant for the attacks we are going to present. Hence, we omit further details and we assume that the inequalities relating t and q in Lemma 1 are respected.

The security of this scheme is based on an analysis from Gentry et al [GHS12], which in turn used parameters presented in the work of Lindner and Peikert [LP11], showing that the scheme is secure as long as the LWE problem parameters d, q, σ obey the inequality

$$d > \log\left(\frac{q}{\sigma}\right)\frac{\lambda + 110}{7.2}.$$

When applied with homomorphic schemes, this relation acquires a challenging aspect. As the standard deviation increases, fewer homomorphic operations can be evaluated, since a larger initial noise would be rapidly propagated. Thus, the ratio q/σ determines the LWE-based cryptography security.

The distribution $\mathcal{D}_{\text{key}}$ must be chosen according to the description of Stehlé and Steinfeld [SS11], such that the public key is close enough to the uniform distribution, so that it reveals almost nothing about the private key. Rigorously, it reveals only a negligible fraction of the secret. Thus, $\mathcal{D}_{\text{key}}$ is a discrete Gaussian on R_q with standard deviation at least $(d\sqrt{\log 8dq})q^k$, for k in the interval $(1/2, 1)$. Furthermore, $\mathcal{D}_{\text{err}}$ is a $\omega(\sqrt{d\log(d)})$-bounded Gaussian distribution. In our attacks we may assume that q is very large in comparison with t and σ_k.

5 Adaptive Key Recovery Attacks

In a key recovery attack, we submit appropriately chosen ciphertexts to a decryption oracle in order to compute the private key. Once the private key is computed, then any ciphertext can later be decrypted. Consequently, a key recovery attack is stronger than a CCA1 attack.

5.1 Attacking the BLLN Scheme for $t > 2$ and Ternary f'

In the original paper [BLLN13], Bos et al stated that we can choose f' and g with coefficients in $\{-1, 0, 1\}$. We call this "ternary f'". We now show that in this case, and when $t > 2$, we can easily compute f' using just one query to the decryption oracle. Recall that $f'[i]$ is the i-th coefficient of the polynomial f'.

Lemma 1. *Let $f = tf' + 1$ where f' has coefficients in $\{-1, 0, 1\}$. Suppose $t \geq 3$ and $6(t^2 + t) < q$. Then,*

$$[\lfloor (t/q)[f[i]\lfloor q/t^2 \rfloor]_q \rceil]_t = f'[i].$$

Proof. Let $\lfloor q/t^2 \rfloor = q/t^2 - \epsilon$ for some $0 \leq \epsilon < 1$. Then,

$$f[i]\lfloor q/t^2 \rfloor = (tf'[i] + 1)(q/t^2 - \epsilon) = f'[i](q/t) + (q/t^2) - \epsilon(tf'[i] + 1)$$

and $[f[i]\lfloor q/t^2 \rfloor]_q = f[i]\lfloor q/t^2 \rfloor - vq$ for some $v \in \mathbb{Z}[x]$. Finally,

$$[\lfloor (t/q)[f[i]\lfloor q/t^2 \rfloor]_q \rceil]_t = [f'[i] + \lfloor 1/t - \epsilon(t^2 f'[i] + t)/q \rceil - vt]_t = [f'[i]]_t$$

since the entries of the polynomial $1/t - \epsilon(t^2 f' + t)/q$ all have absolute value $< 1/3 + 1/6 = 1/2$ (the bound $|t^2 f'[i] + t|/q \leq |t^2 + t|/q < 1/6$ is used here). □

We introduce the informal notation $a \ll b$ to mean that b is much bigger than a (say, $b > 10^6 a$ for parameters in actual cryptosystems). Hence we can observe that $t^2 \ll q$ and so $\lfloor q/t^2 \rfloor$ is a very large integer.

Theorem 1. *Let $t > 2$ and $6(t^2+t) < q$. Let $m_f = \text{DEC}(\lfloor q/t^2 \rfloor)$ be a polynomial in R with coefficients in $[-t/2, t/2]$, where $\lfloor q/t^2 \rfloor$ is a constant integer polynomial that can easily be computed using the public parameters q and t. Then we have that $f = tm_f + 1$.*

Proof. We have that $\text{DEC}(\lfloor q/t^2 \rfloor) = [\lfloor (t/q)[f(\lfloor q/t^2 \rfloor)]_q \rceil]_t$. Because we are multiplying f by a constant polynomial, each coefficient of f is multiplied by $\lfloor q/t^2 \rfloor$. By Lemma 1 we obtain an element in R with coefficients in $\{-1, 0, 1\}$ that equals $f' \in R$. □

Note that the restriction $t > 2$ is a requirement for Lemma 1, but there is also a second reason why it is important. Because $-1 \equiv 1 \pmod 2$, we can't distinguish between -1 and 1 from information modulo 2. Therefore, when $t = 2$ it will be necessary to provide an algorithm to find out the sign of each coefficient.

Algorithm 5.1 uses the ideas described above. We emphasize that the attack is very fast, since it needs to perform just one query to the decryption oracle. Also, the ciphertext that we submit to the decryption oracle is trivial to construct, and the final computation is also very easy.

Algorithm 5.1 BLLN Attack for Ternary Polynomials when $t > 2$ and Ternary f'

Require: The public parameters (q, d, t).
Ensure: The private key f.
$m_f = \text{DEC}(\lfloor q/t^2 \rfloor)$.
return $f = tm_f + 1$.

5.2 Attacking the BLLN Scheme for General f' and $t > 2$

We now consider the case where f' is chosen from $\mathcal{D}_{\text{key}}$ and so has a wider range of possible values. The idea is to make queries on ciphertexts $c_k = \lfloor q/(kt^2) \rfloor$ for various values $k > 1$ to learn information about $[\frac{1}{k} f'] \pmod t$.

Lemma 2. *Let $f = tf' + 1$ where f' is a polynomial whose entries are integers bounded in absolute value by B such that $B^2 < q/(36t^2)$. Let $0 \leq i < d$. Let $k_{\max,i} \leq 2B$ be the maximal integer such that the i-th coefficient of the decryption of ciphertext $\lfloor q/(k_{\max,i}t^2) \rfloor$ is non-zero. Then, we have that, for all $0 \leq i < d$,*

$$|f'[i]| = \lfloor (k_{\max,i} + 1)/2 \rfloor.$$

Proof. The proof is similar to the proof of Lemma 1. Write $c_k = \lfloor q/(kt^2) \rfloor = q/(kt^2) - \epsilon$ for $0 \leq \epsilon < 1$, and note that

$$[fc_k]_q = \tfrac{q}{kt^2}(tf' + 1) - \epsilon(tf' + 1) - vq$$

for some $v \in \mathbb{Z}[x]$. Then,

$$u = \tfrac{t}{q}[fc_k]_q = \tfrac{1}{k} f' + \tfrac{1}{kt} - \epsilon t(tf' + 1)/q - vt$$

is a polynomial with rational coefficients.

We now consider rounding the coefficients of the polynomial $u(x)$ to the nearest integer. For $i > 0$ we have $u[i] = \frac{1}{k} f'[i] - v[i]t$ and so

$$\lfloor u[i] \rceil = \lfloor \tfrac{1}{k} f'[i] \rceil - v[i]t.$$

It follows that the result of the decryption query is $[\lfloor u[i] \rceil]_t = [\lfloor \frac{1}{k} f'[i] \rceil]_t$. Note that if $k > 2B \geq 2|f'[i]|$, then $|\frac{1}{k} f'[i]| < 1/2$ and so the rounded value is zero.

If k is maximal, then $\lfloor \frac{1}{k} f'[i] \rceil \neq 0$ but $\lfloor \frac{1}{k+1} f'[i] \rceil = 0$, and so

$$|\tfrac{1}{k} f'[i]| \geq \tfrac{1}{2} \quad \text{and} \quad |\tfrac{1}{k+1} f'[i]| \leq \tfrac{1}{2}.$$

It follows that

$$\tfrac{k}{2} \leq |f'[i]| \leq \tfrac{k+1}{2}.$$

It remains to deal with the coefficient $f'[0]$, which has an additional error term $\frac{1}{kt} - \epsilon^\star$ where $\epsilon^\star = \epsilon t(tf' + 1)/q$ is added to it. Note that, since $q \gg t(tB + 1)$ and $t > 2$, we have $|\epsilon^\star| \ll 1$. However, we cannot ignore the error as we are adding it to the rational number $\frac{1}{k} f'[0]$. By the same argument as above, we compute

$$|\tfrac{1}{k} f'[0] + \tfrac{1}{kt} - \epsilon^\star| \geq \tfrac{1}{2} \quad \text{and} \quad |\tfrac{1}{k+1} f'[0] + \tfrac{1}{(k+1)t} - \epsilon^\star| \leq \tfrac{1}{2}.$$

It follows that

$$\tfrac{k}{2} \leq |f'[0] + \tfrac{1}{t} - k\epsilon^\star| \quad \text{and} \quad |f'[0] + \tfrac{1}{t} - (k+1)\epsilon^\star| \leq \tfrac{k+1}{2}.$$

Since $(k+1)\epsilon^\star < 3Bt^2 2B/q \leq 1/6$ and $1/t \leq 1/3$ we see there is no rounding error. This completes the proof. □

Note that if $t = 2$ and $k = 1$, then we must be careful about what happens with the independent coefficient, as will be the case in the next section. However, when $t > 2$ we have that if $\lfloor \frac{1}{k} f'[i] \rceil \equiv 1 \pmod{t}$, then $f'[i]$ is positive, while if $\lfloor \frac{1}{k} f'[i] \rceil \equiv -1 \pmod{t}$, then $f'[i]$ is negative, which allows us to completely determine the private key since we know the absolute value and the sign of each coefficient.

The attack is then straightforward. Using binary search and queries to the decryption oracle one can determine $k_{\max,i}$ for $0 \leq i < d$ and hence learn all coefficients. To see that binary search is applicable, note that $|f'[i]| \leq B$ and so $|\frac{1}{2B} f'[i]| \leq 1/2$ and so decryption will generally return 0 for that coefficient. One can then query using $k = B$, and noting that $|\frac{1}{B} f'[i]| \leq 1$ and so the output of decryption is either 0 or ± 1. If the output is ± 1 then $\frac{B}{2} \leq |f'[i]| \leq B$ and one can try $k = (B + 2B)/2 = 3B/2$, while if the output is 0 then $|\frac{1}{B} f'[i]| \leq 1/2$ and one can try $k = B/2$, giving $|\frac{1}{k} f'[i]| \leq 1$, and so on. We give the details as Algorithm 5.2.

Algorithm 5.2 BLLN Attack for General Polynomials when $t > 2$

Require: The public parameters (q, d, t).
Ensure: The private key f.
Let B be the largest possible coefficient of f.
for $i = 1$ till d **do**
Use binary search to find $1 \leq k_{\max,i} \leq 2B$ satisfying the condition of Lemma 2.
$f'[i] = [\text{DEC}(q/(k_{\max,i}t^2))[i].\lfloor (k_{\max,i}+1)/2 \rfloor]_q$.
return $f = tf' + 1$.

The total number of decryption oracle queries, if the algorithm is implemented naively, is $d\lceil \log_2(B) \rceil$. However, this can be improved somewhat by recycling previous oracle values and sub-dividing intervals into t sub-intervals (resulting in $\log_t(B)$ steps in the search) instead of binary splitting and $\log_2(B)$ steps.

5.3 Attacking the BLLN Scheme for $t = 2$

If $t = 2$ we can proceed as in Section 5.2, but our main problem is to find out the sign of each coefficient. Of course, if f is a valid private key then so is $-f$, so we only need to compute f up to a global choice of sign.

Going back to the case of ternary polynomials, we can detect with a single decryption query when the coefficients of f' are zero. But we cannot distinguish when they are 1 or -1, because we are operating modulo 2.

The idea is to make decryption queries to ciphertexts of the form $c = \lfloor q/(t^2k) \rfloor (1 + x^j)$ for suitably chosen k and j. We then get information about $\frac{1}{k}f'(1 + x^j)$. The point is that the i-th coefficient of $f'(1 + x^j)$ is the sum of $f'[i]$ and $f'[i - j \pmod d]$. If the coefficients $f'[i]$ and $f'[i - j]$ are both non-zero then they either cancel to zero or add to ± 2. Hence, taking $k = 2$ we can determine the signs of coefficients relative to each other. By fixing one non-zero coefficient as a "base", we can deduce the sign of all other non-zero coefficients relative to this (as before, we leave the constant coefficient to the end of the algorithm).

When f' is ternary then the details are simple. When f' has general coefficients then the trick is to balance the sizes of coefficients so that cancellation to zero still takes place. So suppose we have run Algorithm 5.2 and determined each coefficient (except perhaps the constant coefficient) $f'[i]$ up to sign. Suppose without loss of generality that $f'[1]$ is non-zero. We will use this as our "base". For each i such that $f'[i]$ is non-zero, we consider the ciphertext

$$c = \lfloor q/(2t^2|f'[1]| \cdot |f'[i]|) \rfloor (|f'[1]| + x^{i-1}|f'[i]|).$$

The i-th coefficient of the decryption of this ciphertext will be

$$\frac{1}{|f'[1]|\cdot|f'[i]|}\left(|f'[1]|\cdot f'[i]+|f'[i]|\cdot f'[1]\right).$$

Hence, if the signs are opposite, then we get a 0 and if the signs are equal, the coefficient is ± 1, which modulo $t=2$ becomes 1. It follows that multiplying the absolute value by the term $(2\text{DEC}(c)-1)$ gives us the desired result.

Algorithm 5.3 BLLN Attack for $t=2$

Require: The absolute value $|f'|$, and the public parameters (q,d).
Ensure: The private key f.
  Run the main part of Algorithm 5.2 to determine $|f'[i]|$ for all $0\leq i<d$.
  Let i_0 be the smallest integer $i>0$ such that $f'[i]\neq 0$.
  $f'[i_0]=|f'[i_0]|$.
  for $i=i_0+1$ till d **do**
    if $|f'[i]|>0$ **then**
      Let $c_{i,i_0}=\lfloor q/(2t^2|f'[i_0]|\cdot|f'[i]|)\rfloor(|f'[i_0]|+x^{i-i_0}|f'[i]|)$.
      $f'[i]=(2.\text{DEC}(c_{i,i_0})[i]-1).|f'[i]|$.
  Find three candidate values for $f'[0]$ and test the three possible values for f using h
  return $f=tf'+1$.

Therefore, after calling algorithm 5.2, we must use algorithm 5.3 to determine the sign of each coefficient of the private key. But we still have to solve the problem of the independent coefficient, mentioned in last section. As we have seen, the term $1/t-\epsilon^\star$ can change the result of rounding to the nearest integer. For instance, considering the case of ternary f' and $t=2$, then we have that $k=1$ and in the case that $f'[0]=-1$, we have that

$$[\lfloor -1+1/2+\epsilon^\star\rceil]_t=0$$

and the decryption oracle returns 0 instead of 1 as expected. Then we have to distinguish between two cases: $f'[0]=-1$ and $f'[0]=0$. But since we have arbitrarily chosen the sign of $f[i_0]$ as positive, then we must check also the case $f[0]=1$. Hence we have three candidates for f'. We can check which of them satisfies the requirement that $(tf'+1)h$ in R_q is a polynomial with small coefficients. This completes the attack.

There are at most $d-1$ additional decryption oracle queries to determine the sign.

5.4 Attacking the LTV Scheme

In this section we assume that q is odd. The LTV scheme is extremely similar to the BLLN scheme. The two schemes are based on the same algebraic structure, and the key generation algorithms are essentially the same, with the only difference that LTV is restricted to the case $t = 2$. The LTV scheme is not scale-invariant, leading to simpler algorithms. Our focus is the decryption algorithm, so we explain this now.

Decryption. Compute $m = [fc]_q$. Output $m \pmod 2$.

The paper [LATV12] is vague about the exact computation of the decryption algorithm. The value m is a polynomial in R_q with small coefficients, so it is natural to interpret it as an element of $R = \mathbb{Z}[x]/(x^d + 1)$. The ambiguity comes in the next step. Does $m \pmod 2$ mean only the constant term of the polynomial modulo 2, or the whole polynomial reduced modulo 2? In our attack we assume the latter case. The former case can be reduced to the latter case by replacing a decryption query on c by d decryption queries on cx^i for $0 \leq i < d$.

The attack is therefore seen to be more-or-less identical to the attack in the previous section. Let $k \geq 1$ be an integer and consider the ciphertext $c_k = 2\lfloor q/(4k) \rfloor$. Lemma 3 shows why we can compute f' using the same strategy as before.

Lemma 3. *Let $c_k = 2\lfloor q/(4k) \rfloor$. Let $k^\star_{\max,i}$ be the maximal integer such that $\textsc{Dec}(c_{k^\star_{\max,i}})[i]$ is non-zero. Then we have that $f[i]$ is given by $k^\star_{\max,i} + 1$.*

Proof. First note that c_k is an even integer and so $(2f' + 1)c_k$ is an integer polynomial with even coefficients.

For $k \geq 1$ we have that $c_k = q/(2k) - \epsilon$ for some $0 \leq \epsilon < 2$, and decryption of c_k first computes

$$(2f' + 1)c_k = f'(q/k - 2\epsilon) + 2\lfloor q/(4k) \rfloor.$$

Note that $q/k - 2\epsilon$ is an even integer. Thus, if k is big when compared to $f'[i]$, reduction by q does not change the value, then after reducing by 2 we get zero. If $f'[i] \geq k$ then $f'[i](q/k) \geq q$ and so, as long as the error term is small enough, $f'[i](q/k - 2\epsilon) - q$ is odd. It follows that $[fc_k]_q \pmod 2$ is odd and so the condition $f'[i] > k$ can be tested using a decryption oracle query. Hence, we proceed using the same method as before. One chooses maximal $k^\star_{\max,i}$ such that $f'[i] > k^\star_{\max,i}$ and hence determines the value of $|f'[i]|$. For instance, we have that $|f'[i]| = k^\star_{\max,i} + 1$. The signs and the independent coefficient are handled in the same way as above.

□

Algorithm 5.4 LTV Attack

Require: The public parameters (q, d, t).
Ensure: The absolute value of the private key f.
Let B be the largest possible coefficient of f.
for $i = 1$ till d **do**
Use binary search to find $1 \leq k^{\star}_{\max,i} \leq 2B$ satisfying the condition of Lemma 3.
$|f[i]| = [(k^{\star}_{\max,i} + 1)]_q$.
return f.

6 Concluding Remarks

We have described adaptive key recovery attacks on NTRU-based SHE schemes. Other families of SHE schemes, as represented in Figure 1, are also vulnerable to this kind of attack, showing that CCA1 security is hard to achieve in homomorphic encryption. Adaptive key recovery attacks on homomorphic encryption seem to be realistic in certain scenarios, so they are potentially a serious problem in practice. The only homomorphic encryption scheme known to resist such attacks is the scheme by Loftus et al [LMSV12].

Acknowledgements

We thank Qiang Tang and the anonymous referees for helpful comments.

References

BGV12. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) Fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 309–325, New York, NY, USA, 2012. ACM.

BLLN13. J. W. Bos, K. Lauter, J. Loftus, and M. Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In Martijn Stam, editor, *IMA Int. Conf.*, volume 8308 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2013.

BLN14. J. W. Bos, K. Lauter, and M. Naehrig. Private predictive analysis on encrypted medical data. *Journal of Biomedical Informatics*, 50:234–243, 2014.

Bra12. Z. Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In *Advances in Cryptology - Crypto 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886. Springer, 2012.

BV11a. Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 97–106, Washington, DC, USA, 2011. IEEE Computer Society.

BV11b. Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *Proceedings of the 31st Annual Conference on Advances in Cryptology*, CRYPTO'11, pages 505–524, Berlin, Heidelberg, 2011. Springer-Verlag.

BV14. Z. Brakerski and V. Vaikuntanathan. Lattice-based FHE as secure as PKE. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, ITCS '14, pages 1–12, New York, NY, USA, 2014. ACM.

CCK+13. J. Cheon, J. Coron, J. Kim, M. Lee, T. Lepoint, M. Tibouchi, and A. Yun. Batch fully homomorphic encryption over the integers. In T. Johansson and P. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 315–335. Springer Berlin Heidelberg, 2013.

CMNT11. J. Coron, A. Mandal, D. Naccache, and M. Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In *Proceedings of the 31st annual conference on Advances in cryptology*, CRYPTO'11, pages 487–504, Berlin, Heidelberg, 2011. Springer-Verlag.

CNT12. J. Coron, D. Naccache, and M. Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 446–464. Springer Berlin Heidelberg, 2012.

CT14. M. Chenal and Q. Tang. On key recovery attacks against existing somewhat homomorphic encryption schemes. In *Latincrypt (to appear)*, Florianópolis-SC, Brazil, 2014.

DHS14. Y. Doröz, Y. Hu, and B. Sunar. Homomorphic AES evaluation using NTRU. Cryptology ePrint Archive, Report 2014/039, 2014. `http://eprint.iacr.org/`.

Gen09a. C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`.

Gen09b. C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 169–178, New York, NY, USA, 2009. ACM.

GHS12. C. Gentry, S. Halevi, and N. P. Smart. Homomorphic evaluation of the AES circuit. In R. Safavi-Naini and R. Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 850–867. Springer Berlin Heidelberg, 2012.

GS11. C. Gentry and Halevi S. Implementing gentry's fully-homomorphic encryption scheme. In Kenneth Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 129–148. Springer Berlin / Heidelberg, 2011.

GSW13. C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92. Springer Berlin Heidelberg, 2013.

HPS98. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In J. P. Buhler, editor, *Algorithmic Number Theory*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer Berlin Heidelberg, 1998.

LATV12. A. López-Alt, E. Tromer, and V. Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 1219–1234, New York, NY, USA, 2012. ACM.

LLAN14. K. Lauter, A. Lopez-Alt, and M. Naehrig. Private computation on encrypted genomic data. Technical Report MSR-TR-2014-93, June 2014.

LMSV12. J. Loftus, A. May, N. P. Smart, and F. Vercauteren. On CCA-secure somewhat homomorphic encryption. In A. Miri and S. Vaudenay, editors, *Selected Areas in Cryptography*, volume 7118 of *Lecture Notes in Computer Science*, pages 55–72. Springer Berlin Heidelberg, 2012.

LP11. R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In *Proceedings of the 11th International Conference on Topics in Cryptology*, CT-RSA'11, pages 319–339, Berlin, Heidelberg, 2011. Springer-Verlag.

LPR13. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. *J. ACM*, 60(6):43, 2013.

RAD78. R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation, Academia Press*, pages 169–179, 1978.

SS10. D. Stehlé and R. Steinfeld. Faster fully homomorphic encryption. In M. Abe, editor, *Advances in Cryptology - ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 377–394. Springer Berlin Heidelberg, 2010.

SS11. D. Stehlé and R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, pages 27–47, 2011.

SV10. N. P. Smart and F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In P. Q. Nguyen and D. Pointcheval, editors, *Public Key Cryptography – PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer Berlin Heidelberg, 2010.

vDGHV10. M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *Proceedings of the 29th Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'10, pages 24–43, Berlin, Heidelberg, 2010. Springer-Verlag.

ZPS12. Z. Zhang, T. Plantard, and W. Susilo. On the CCA-1 security of somewhat homomorphic encryption over the integers. In *Proceedings of the 8th International Conference on Information Security Practice and Experience*, ISPEC'12, pages 353–368, Berlin, Heidelberg, 2012. Springer-Verlag.

2.3 Using verifiable computation to avoid the attacks

In this section we present a method that can be used to fix the problem raised in last section, namely the possibility of applying key-recovery attacks to homomorphic encryption schemes.

In order to do that, we use ideas proposed by Fiore, Gennaro and Pastro [39], to construct a verifiable computation scheme for quadractic multivariate polynomials. Intuitively, if it is possible to verify the correctness of homomorphic computation, then it is hard for an adversary, with non-negligible probability, to come up with interesting decryption queries, in the sense that only queries that were output by the encryption oracle, or by a valid homomorphic computation are possible. Therefore, such queries cannot be used to attack the scheme because an adversary would have to know the message in advance, before submitting its corresponding ciphertext to the decryption oracle.

This paper was submitted to the International Journal of Applied Cryptography.

AGCD-based CCA1-secure somewhat homomorphic encryption using verifiable computation

Abstract: We construct a practical CCA1-secure somewhat homomorphic encryption scheme combining verifiable computation and homomorphic encryption over the integers. The scheme prevents key recovery attacks in homomorphic encryption systems, which are relevant in the cloud computing scenario where a decryption oracle can be obtained by monitoring the client behavior for certain messages. The intuition behind the construction comes from simulating the decryption oracle in the verifiable computation scheme described by Fiore, Gennaro and Pastro [FGP14] to obtain an encryption scheme for quadratic multivariate polynomials. In that work, Fiore *et al.* employed the BGV [BGV12] cryptosystem as the underlying homomorphic encryption scheme. We show that using the DGHV scheme [vDGHV10] and the Chinese Remainder Theorem to allow batch operations [CCK$^+$13], it is possible to achieve a better overhead when compared to the BGV scheme. The scheme described in this paper can be used to homomorphically calculate statistical functions in such an way that the computation can be efficiently verified by the client using bilinear pairings.

Keywords: CCA1-security; Somewhat Homomorphic Encryption; Verifiable Computation; Key Recovery Attacks.

1 Introduction

Homomorphic encryption offers an interesting solution to many cloud computing security problems. Although fully homomorphic encryption is not yet practical, somewhat homomorphic encryption can be used to solve relevant problems under some restrictions in a privacy-preserving way [NLV11]. However, almost every scheme in the literature is secure only against *chosen plaintext attacks* (CPA) and such proposals are vulnerable against key recovery attacks [DGM15]. In summary, the problem is that the private key can be computed if there is access to a decryption oracle. The only scheme secure against *chosen ciphertext attacks* (CCA) is the one proposed by Loftus et al. [LMSV12], and thus alternate constructions are important to have. In this scheme, the algebraic properties allow one to verify the validity of ciphertexts, and it is not viable for an adversary to derive a valid ciphertext in a way other than honestly using the encryption algorithm or using homomorphic operations over known valid ciphertexts. In other words, they prove that the construction has a property related to *plaintext awareness* which is then used to prove that the cryptosystem is CCA1-secure.

A possible solution to the problem above is to construct a CCA2-secure *conventional* secret key cryptosystem, where conventional means the encryption scheme is not homomorphic. In order to accomplish this task, a standard strategy is to employ a secure MAC construction to authenticate ciphertexts, considering that the CCA adversary will explore access to the decryption oracle to break the scheme. By authenticating ciphertexts, we avoid undesirable decryption queries for ciphertexts that have not been computed regularly by the encryption algorithm. Namely, it is possible to construct a *CCA2-secure conventional* secret key (non-homomorphic) cryptosystem using a CPA-secure encryption system and MACs. This kind of construction is known as *Encrypt-then-MAC* pradigm [KL07]. By the security definition of MAC, an adversary cannot forge an authentication tag with non-negligible probability. Thus it is not viable to generate a valid ciphertext without the help of an encryption oracle. In other words, since the adversary cannot forge MACs, he must know beforehand the message corresponding to any ciphertext, making the decryption oracle useless. In the security experiment, the simulator can store encryption queries and simulate the decryption oracle by checking if the authentication tag is valid, returning the plaintext that was previously queried. Otherwise, the simulator returns the *invalid cphertext* symbol.

Recently, many *homomorphic MAC* schemes have been proposed [CFGN15, CF13, GW13]. A natural application of such schemes is to verify the validity of ciphertexts using the Encrypt-then-MAC paradigm and hence this strategy could also be used to solve the aforementioned problem. Unfortunately, the strategy cannot be used with homomorphic encryption, because given a set of known ciphertexts, new valid ciphertexts can easily be obtained by combining them homomorphically. Thus it is not hard for an adversary to compute valid ciphetexts that could be submitted to a decryption oracle in order to attack the scheme. For example, in the context of homomorphic encryption over the integers, the binary GCD attack works by submitting a sequence of ciphertexts in the form $(c_1 - c_2)/2$, where each c_i is a valid ciphertext. These submissions

Copyright © 2016 Inderscience Enterprises Ltd.

Copyright © 2016 Inderscience Enterprises Ltd.

are perfectly valid because they are the result of a homomorphic computation, thus protecting against such kind of attack is a challenging problem.

To circumvent the problem described above one could use a secure verifiable computation ($\mathcal{VC}$) scheme, as for example the one described by Fiore, Gennaro and Pastro [FGP14]. This proposal, called FGP from now on, achieves an *amortized cost* to verify that an authentication tag corresponds to a certain computation. This allows the verification of ciphertext validity to be faster than repeating the entire computation again, considering that it was obtained by homomorphically evaluating a given function. The key elements to achieve this goal are an amortized PRF function and a homomorphic hash function that allows to considerably reduce the size of the ciphertexts.

This paper tackles the problem of constructing a practical CCA1-secure somewhat homomorphic encryption scheme and reaches a positive answer. The proposed scheme employs the properties of adaptive security and privacy from the FGP construction to make hard for an adversary to come up with valid ciphertexts without using the encryption algorithm or having knowledge of the secret key. The FGP cryptosystem depends on using a secret to generate authentication tags and also to verify if the computation was carried out correctly. Hence, when the scheme is used together with a public key CPA-secure homomorphic encryption scheme, we lose the public key property. We could fix this problem by publishing (authenticated) encryptions of zero, similarly to Rothblum's method [Rot10], but first of all we would need to provide a mechanism to verify if the subjacent encryption algorithm outputs ciphertexts that were correctly computed. Instead, because we are interested in the cloud computing scenario, it makes sense to use a symmetric cryptosystem, since the cloud requires only an evaluation key in order to homomorphically compute over ciphertexts, which are provided by the client. Such strategy not only allows us to verify if the cloud is computing exactly what the user wants, but also provides a homomorphic cryptographic solution that avoids CCA1 attacks.

Nevertheless, the previously mentioned ($\mathcal{VC}$) construction requires bilinear pairings, imposing a big restriction on the class of functions that can be verified. Such functions can only compute one multiplication, restricting the class of functions to quadratic multivariate polynomials. Hence, the motivation of this paper is to use homomorphic encryption together with verifiable computation to avoid key recovery attacks. However, since the FGP construction can only deal with one multiplication, an interesting problem would be to extend this scheme to higher degree polynomials.

Finally, since the FGP scheme uses the BGV scheme, it can perform batch processing through the Chinese Remainder Theorem, what allows to reduce the overhead introduced by bilinear pairings. Therefore, because of the number of slots that the scheme can handle in parallel, the pairing computation overhead is amortized by this parallelism. On the other hand, we have not much flexibity in the choice of the number of slots, what may be a concern in order to adapt the technique for some applications. Another important issue is that parallel computation is not a complete set of operations, unless it is possible to permute the slots. Even though the BGV scheme allows to permute the slots, the FGP scheme does not describe how to do verifiable computation of these permutations, and this issue is an interesting question.

1.1 Notation

Notation $\lfloor a \rceil$ is used to round a to the nearest integer, while notation $[a]_q$ is used to denote centralized modular reduction, i.e. reduction modulo q, but with result given in the interval $(-q/2, q/2]$. If a is a polynomial, $[a]_q$ is the centralized modular reduction of each coefficient of a (analogously for $\lfloor a \rceil$). We use $[a_i]$ to denote a sequence of indexed variables a_i, where the index range is clear from the context.

1.2 Contributions

The main contributions of the paper are enumerated as follows:

1. We extend the construction of verification computation to homomorphic encryption over the integers. For instance, we show how to compute the universal hash function in order to get a homomorphic collision resistant hash function based on the AGCD problem. Furthermore, we explicitly describe the homomorphic MAC scheme that is used to construct the verifiable computation scheme in [FGP14], what makes our description better organized and thus simpler to follow;

2. We show that the proposed scheme achieves a better overhead when compared to the BGV scheme, where the overhead is measured as the ratio between the ciphertext size and the plaintext size;

3. We show how to simulate the decryption oracle in the security proof of the verifiable computation scheme described by Fiore, Gennaro and Pastro [FGP14]. Hence, the scheme can be proven to be CCA1-secure. Moreover, it allows verification queries and thus is resistant against CVA attacks, what means that the scheme achieves the highest security level possible for homomorphic encryption.

1.3 Organization

In Section 2 we give the basic definitions and the security model under which we are going to work. In Section 3 we instantiate the scheme and describe how to use verifiable computation together with homomorphic

encryption over the integers to obtain a CVA-secure scheme that can deal with multiplicative depth equal to one. In Section 4 we discuss the results obtained and compare with previous work. In Section 5 we describe related work. In Section 6 we describe some possible applications and in Section 7 we give the final remarks and conclusions.

2 Definitions and security model

In this section we are going to define the cryptographic primitives used later, and for each primitive, we give the adopted security model.

2.1 Homomorphic encryption

In 2009, Gentry [Gen09] proposed the first construction of *fully homomorphic encryption* (FHE), solving an important open problem in cryptography, that was conjectured in 1978 by Rivest, Adleman and Dertouzos [RAD78]. Ciphertexts produced by an FHE scheme can be added or multiplied, in such a way that we obtain the corresponding addition or multiplication of the respective plaintexts. The ability to algebraically operate over ciphertexts is of great importance because we can transform any algorithm into a sequence of additions and multiplications. Therefore, such a scheme can be used to provide security in the context of cloud computing, because the cloud can evaluate any algorithm solely with access to the encryption of its input, and such that the computation returns the encryption of the output.

Definition 2.1: Let λ represent the security parameter of the cryptosystem. A secret key homomorphic encryption scheme $\mathcal{E}_{\text{CPA}}$ is defined by the algorithms:

$$(\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Eval}).$$

Key generation. Given the security parameter λ in unary representation, algorithm $\text{KeyGen}(1^\lambda, f)$ generates the secret key $\mathtt{dk}$, which is used to encrypt and decrypt ciphertexts, and an evaluation key $\mathtt{edk}$ that can be used to do homomorphic operations. Formally, it is important to remark that the function f is implemented by an algebraic circuit over a certain ring, i.e. a sequence of additions and multiplications of elements taken from this ring. Hence, using f to make reference to the *description* of the function is an abuse of notation and we adopt this choice from now on in order to simplify our definitions.

Encryption. We denote by $\mathcal{M}$ and $\mathcal{C}$ the plaintext and the ciphertext space, respectively. Given $m \in \mathcal{M}$, the algorithm Enc returns $c = \text{Enc}_{\mathtt{dk}}(m)$, where $c \in \mathcal{C}$.

Decryption. Given $c \in \mathcal{C}$, the algorithm Dec ouputs either $m = \text{Dec}_{\mathtt{dk}}(c)$ or the invalig tag $\perp$ if the ciphertext c is not correctly formed.

Evaluation. Given a function f whose inputs are the plaintexts $m_1, \ldots, m_t$, for $m_i \in \mathcal{M}$, the Eval algorithm computes the ciphertext $c \in \mathcal{C}$, such that $c = \text{Eval}_{\mathtt{edk}}([c_i], f)$ and $c_i = \text{Enc}_{\mathtt{dk}}(m_i)$ for every $0 \leq i \leq t$.

Definition 2.2: Given $c = \text{Dec}_{\mathtt{dk}}(m)$, we say that the encryption scheme $\mathcal{E}_{\text{CPA}}$ has *encryption correctness* if $m = \text{Enc}_{\mathtt{dk}}(c)$. Furthermore, the *homomorphic correctness* is given by the following condition:

$$\text{Dec}_{\mathtt{dk}}(\text{Eval}_{\mathtt{edk}}([c_i], f)) = f([m_i]).$$

Definition 2.3: A cryptosystem is secure against *chosen ciphertext attack* (CCA2) if there is no polynomial time adversary $\mathcal{A}$ that can win the following game with non-negligible probability.

Setup. The challenger obtains $(\mathtt{dk}, \mathtt{edk}) = \text{KeyGen}(1^\lambda, f)$ and sends $\mathtt{edk}$ to adversary $\mathcal{A}$.

Decryption queries. $\mathcal{A}$ sends ciphertexts to the challenger, who returns the corresponding plaintexts.

Verification queries. $\mathcal{A}$ sends ciphertexts to the challenger and, for each ciphertext, the challenger returns a bit $v = 0$ to the adversary if the ciphertext is invalid and returns the bit $v = 1$ if the ciphertext is valid.

Challenge. The adversary randomly generates two plaintexts $m_0, m_1 \in \mathcal{M}$ and sends them to the challenger, who chooses randomly a bit $b \in \{0, 1\}$ and computes the ciphertext $c = \text{Enc}_{\mathtt{dk}}(m_b)$. The challenger sends c to $\mathcal{A}$.

Answer. $\mathcal{A}$ sends a bit b' to the challenger and wins the game if $b' = b$.

If we do not allow verificaton queries, but do allow decryption queries only before the challenge, we say that the cryptosystem is secure against CCA1 adversaries (lunchtime attacks). If instead we only allow access to an encryption oracle, i.e., the adversary can choose any message that is distinct from m_0 and m_1 to be encrypted under the same key pair, then we say that the cryptosystem is secure against *chosen plaintext attacks* (CPA). If verification queries are allowed during the entire experiment, but after the challenge we do not allow decryption queries anymore, we say that the scheme is secure against *chosen verification attacks* (CVA).

In homomorphic encryption, it is impossible to achieve CCA2 security, because the adversary can add an encryption of zero to the encrypted challenge, or multiply it by the encryption of one, and send it to the decryption oracle, which allows him to trivially win the game. Many FHE schemes have as public value an encryption of the private key bits, which can be sent to the decryption oracle before the challenge, which makes such schemes insecure against CCA1 adversaries. Indeed, a *key recovery* attack is stronger than a CCA1 attack and Loftus et al [LMSV12] showed that Gentry's construction over ideal lattices is vulnerable to it and presented the only SHE proposal that is known to be CCA1-secure. In the same work, the authors show that the proposed scheme is not CVA-secure. Hence the

construction of a CVA-secure encryption scheme is still an open problem.

2.2 Homomorphic MAC

MACs are useful to *authenticate* data. *Homomorphic MACs* thus can be used to combine authentication tags in order to authenticate the evaluation of a certain function over data. A trivial solution would be to send not only the output, but also the entire input to the function, in such an way that one could check if the computation was carried out correctly. However, this solution is obviously not desirable. Hence, a minimum requirement for a candidate solution would be *succintness*, which informally can be described as the capacity of obtaining a communication complexity that is significantly less than sending the input data to the receiver. Recently, Gennaro and Wichs [GW13] proposed the construction of *fully homomorphic MACs*, which accomplishes this minimum requirement, but still fails to be a practical solution. Catalano and Fiore [CF13] proposed a practical construction that gives us a tradeoff between succintness and *composability*, which is the ability to compose the output of different computations.

2.2.1 Multi-labels

Let f be a function on t plaintexts $m_1, \ldots, m_t$. We want to use homomorphic encryption to compute a ciphertext c that corresponds to an encryption of $m = f([m_i])$. However, if we outsource the computation of f to the cloud, where the computation is done by homomorphically evaluating f over ciphertexts $c_i = \textsc{Enc}(m_i)$, it would be important to be able to verify if the cloud performed exactly the computation we want and, in order to accomplish this task, we can associate each input of f with a *label*, as in the homomorphic MAC scheme proposed by Catalano and Fiore [CF13]. However, in their construction we can not reuse labels, what would immediately imply that we can compute f only once. To solve this problem, Backes, Fiore and Reischuk [BFR13] proposed the concept of *multi-labels*. The purpose of using multi-labels is for identifying a certain input of f in each time that the function is computed. A multi-label is defined by the pair (Δ, i), where Δ identifies the data set for each time f is evaluated and i identifies the index $1 \leq i \leq t$ of the input.

We can make an analogy with a table with t columns that stores in each row a data set that will be used as input to evaluate the function f. Different rows represent distinct evaluations of f and are identified with values $\Delta_1, \Delta_2, \cdots$ that are mutually distinct. For example, consider that f is an statistical function over stock exchange prices. We can imagine that Δ_j represents the company name and the index i represents each instant of time along some determined period. Thus multi-labels can be used to organize the information in such way that it is possible to isolate different categories of data. This characterist is indeed desirable and helps to provide security to the outsorced computation solution, as we are going to detail later.

Next we define homomorphic MACs. In particular, we remark that the given definition authenticates ciphertexts instead of plaintexts, since our goal is to use it later in the verifiable computation scheme.

Definition 2.4: A homomorphic MAC scheme HomMAC is given by the algorithms: $(\textsc{KeyGen}, \textsc{Auth}, \textsc{Ver}, \textsc{Eval})$, defined as follows:

Key generation. Given the security parameter λ and the description of the function f, the algorithm $\textsc{KeyGen}(1^\lambda, f)$ generates a secret key $\mathtt{vk}$ that will be used to authenticate and verify tags, and an evaluation key $\mathtt{evk}$ that will be used to compute the evaluation of a function f over authentication tags.

Authentication. Given a ciphertext $c_i \in \mathcal{C}$, corresponding to the multi-label (Δ, i), the algorithm $\textsc{Auth}$ computes $\sigma = \textsc{Auth}_{\mathtt{vk}}(c, \Delta, i)$.

Verification. The verification algorithm $\textsc{Ver}$ receives $\sigma, c, \Delta, i, f^\star$ and $\mathtt{vk}$ as input and returns a bit 1 for valid authentication tags and returns 0 otherwise. The function $f^\star$ is either the function f given as input to the $\textsc{KeyGen}$ algorithm or the identity function f_{ID}, which output is its own input. Formally, we have that

$$\Pr[\textsc{Ver}_{\mathtt{vk}}(\sigma, c, \Delta, i, f^\star) = 1)] = 1.$$

Evaluation. Given the evaluation key $\mathtt{evk}$, and the authentication tags σ_i, for $0 \leq i \leq t$, the $\textsc{Eval}$ algorithm returns another authentication tag σ which corresponds to the evaluation of the function f over $[\sigma_i]$. Specifically, we have that $\sigma = \textsc{Eval}_{\mathtt{evk}}([\sigma_i], f)$, $\sigma_i = \textsc{Auth}_{\mathtt{vk}}(c_i, \Delta, i)$ for each $0 \leq i \leq t$, $c = \textsc{Eval}_{\mathtt{edk}}([c_i], f)$ and $\textsc{Ver}_{\mathtt{vk}}(\sigma, c, \Delta, i, f)$ return 1.

Definition 2.5: The security is defined according to the following experiment between a challenger and a polynomial time adversary $\mathcal{A}$.

Setup. The challenger runs $(\mathtt{vk}, \mathtt{evk}) = \textsc{KeyGen}(1^\lambda, f)$ and initializes the list T with the empty set.

Authentication queries. The adversary asks for the authentication of a ciphertext c_i and the corresponding multi-label (Δ, i). The challenger verifies the list T to know if the multi-label was previously used. In other words, it verifies if $[\Delta, i, c, \cdot] \in T$. If it is the case, the challenger returns the same authentication tag σ_i that was previously computed. If (Δ, i) is in the list T, but with a different ciphertext c_i', then the challenger ignores the query. Otherwise, the challenger computes $\sigma = \textsc{Auth}(c_i, \Delta, i, \mathtt{vk})$, returns σ_i to the adversary and stores $[\Delta, i, c_i, \sigma]$ in T.

Verification queries. The adversary $\mathcal{A}$ submits queries (σ, c, f) to the challenger, which replies with $\textsc{Ver}_{\mathtt{vk}}(\sigma, c, \Delta, i, f)$.

Forgery. The adversary $\mathcal{A}$ outputs $(\sigma^\star, c^\star, f)$.

The function f is called *well-defined* if no input, which index belongs to the interval $0 \leq i \leq t$, is never

used during its evaluation. In other words, there is no useless index i. Also, we say that f is *well-defined* with respect to the list T if for each $\Delta \in T$ and for every index $0 \leq i \leq t$ we have that $(\Delta, i) \in T$. We call the forgery *type 1*, if f is not well defined on T and *type 2* if it is well defined on T and $c^\star$ is not the correct output of f, evaluated over previously authenticated inputs.

A homomorphic MAC scheme HomMAC is secure if no adversary can produce a forge in the above experiment with non-negligible probability. It is important to remark that multi-labels can not be reused by definition. Hence, an adversary cannot submit multiple queries using the same multi-label in order to get information that allows $\mathcal{A}$ to output a forge.

2.3 Homomorphic verifiable computation

Although homomorphic encryption gives us a very flexible cryptographic primitive, when applied to the cloud computing scenario, it lacks an important property: the ability to verify if homomorphic computation corresponds to what the client desires. A verifiable computation scheme could solve this problem and we have two requirements that such a scheme must respect. First, the cloud must not take much more time to do the verifiable computation when compared to the non-verifiable solution. Second, the client must be able to verify the result faster than doing the entire compution by himself. Some proposals [GGP10, CKV10] use homomorphic encryption to construct a $\mathcal{VC}$ scheme, what is interesting, because it is possible to offer input and output privacy, since both are encrypted. However, the underlying security model does not allow verification queries. Recently, Fiore, Gennaro and Pastro [FGP14] proposed a new construction that does allow verification queries, improving the security model. They showed how to solve practical problems as for example the proposal of a $\mathcal{VC}$ scheme to compute quadratic multivariate polynomials over encrypted data, which can be used to homomorphically compute statistical functions. We remark that this application requires only one level of multiplications, what is an important characteristic to be considered to calculate the parameters of the underlying SHE scheme.

Definition 2.6: A verifiable computation scheme $\mathcal{VC}$ is defined by the algorithms $(\textsc{KeyGen}, \textsc{ProbGen}, \textsc{Compute}, \textsc{Verify})$, as follows:

Key generation. The algorithm $\textsc{KeyGen}(1^\lambda, f)$ generates secret key $\mathtt{sk}$ and an evaluation key $\mathtt{esk}$.

Problem generation. Using the secret key $\mathtt{sk}$, the algorithm $\textsc{ProbGen}$ receives as input a ciphertext c_i and computes the corresponding authentication tag σ_i such that $\sigma_i = \textsc{Auth}_{\mathtt{vk}}(c_i, (\cdot, i))$.

Verification. Given the secret $\mathtt{sk}$, σ and the ciphertext c, we have that $\textsc{Verify}_{\mathtt{sk}}(\sigma, c)$ returns 1 if $c = f([c_i])$ and $\sigma = \textsc{Auth}_{\mathtt{vk}}(c, (\cdot, i))$. Otherwise it returns 0.

Evaluation. Given $\sigma_1, \ldots, \sigma_t$ and the description of function f, the algorithm $\textsc{Compute}_{\mathtt{esk}}([\sigma_i], \Delta, f)$ returns the authentication tag σ that corresponds to the ciphertext $c = \textsc{Eval}_{\mathtt{edk}}([c_i], f)$, obtained by running the $\textsc{Eval}$ algorithm from the underlying homomorphic encryption scheme. We say that the $\mathcal{VC}$ scheme is *correct* if $\textsc{Verify}_{\mathtt{sk}}(\sigma, c)$ outputs 1.

Definition 2.7: Consider a polynomial time adversary $\mathcal{A}$ that can make the following queries to an oracle that has setup the experiment after obtaining a valid key pair by running the $\textsc{KeyGen}$ algorithm.

Problem generation queries. The adversary chooses a ciphertext c_i and a multi-label (Δ, i) to submit to the oracle, that returns the corresponding authentication tag $\sigma_i = \textsc{Auth}_{\mathtt{vk}}(c, \Delta, i)$.

Verification queries. The adversary chooses a ciphertext c and an authentication tag σ to submit to the oracle, that returns the acceptance bit $v \in \{0, 1\}$.

After a polynomial number of queries, $\mathcal{A}$ produces a forge to the $\mathcal{VC}$ scheme, as follows.

Forge computation. The adversary generates an authentication tag $\hat{\sigma}$ and a ciphertext $\hat{c}$. We say that $\mathcal{A}$ succeeds if $\textsc{Verify}_{\mathtt{sk}}(\hat{\sigma}, \hat{c})$ returns 1.

The $\mathcal{VC}$ scheme is secure if $\mathcal{A}$ can not produce a computation forge with non-negligible probability.

3 Cryptographic primitives

3.1 Bilinear maps

As we are going to describe later, the construction of homomorphic hash and amortized closed-form PRF will be based on the existence of a secure asymmetric bilinear map $\mathbf{bp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e, g, h)$, where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_t$ are groups of sufficiently large prime order, g and h are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively and e is an appropriate choice of bilinear map, satisfying the usual requirements: (i) non-degeneracy; (ii) efficiently computable and (iii) bilinearity. This cryptographic primitive will be used as a black-box in the next sections and it is important to remark that care must be taken when instantiating such primitive [GPS08, KU16]. An efficient implementation is offered by the RELIC library [AG].

3.2 Homomorphic encryption

In this section we are going to describe how to construct homomorphic encryption over the integers, which corresponds to a family of homomorphic encryption schemes which security is based on the *approximate greatest common divisor* (AGCD) problem. Other two families, namely the BGV and NTRU schemes, can also be considered as an interesting option to construct practical somewhat homomorphic cryptosystems. Initial proposals for the first family, although simpler, have considerably worse performance. However, many optimizations were proposed in the literature and, in recent work, Cheon and Stehlé [CS15a] proved that

the underlying hard problem for BGV and NTRU constructions, namely the LWE problem, can be reduced to a particular instantiation of the AGCD problem. They also presented a reduction in the reverse direction, showing that both problems are in some sense equivalent. Furthermore, using the tensor technique from the BGV construction, it is possible to construct an AGCD-based scheme with better parameters when compared to previous AGCD-based proposals. Nevertheless, it is not yet clear how to extend their ideas to be able to do batch operations using the Chinese Remainder Theorem as we are going to describe. Moreover, since we want to achieve multiplicative depth only equal to one, the tensor technique is not better than the conventional method, because although the multiplicative noise grows only additively, it has an inherent constant factor that is proportional to $\log \gamma^2$, where γ is the ciphertext size.

When naively considered, the NTRU-based family seems to have better performance than the BGV-based family, but the analysis of which scheme is better depends on many circunstances. For instance, it is important to consider the multiplicative depth of the homomorphic computation, the plaintext size and the utilization of batch operations by the Chinese Remainder Theorem. Ana Costache and Nigel Smart [CS15b] compared BGV and NTRU with respect to these criterias, concluding that BGV is better for bigger plaintext size, while NTRU is the best option for small plaintext size. However, to accomplish relatively large multiplicative depth than we are interested in, the authors employ techniques such as Modulus Switching and Key Switching. Indeed, we intend to achieve multiplicative depth one and this characteristic permits to simplify the constructions by avoiding such techniques.

Another comparison between BGV and the *scale invariant* NTRU was done by Miran Kim and Kristin Lauter [KL15]. The authors conclude that, for low multiplicative depth and small plaintext slot, NTRU may offer an advantage against BGV. Again, since we are interested in multiplicative depth of one, but large plaintext slot, we do not need an scale invariant scheme and for simplicity this concept will not be considered. Finally, because we are going to encode large numbers inside each slot, the works mentioned above provide evidence that NTRU is not a good option and we are going to focus only on the comparison between the BGV scheme and the AGCD-based construction.

3.2.1 AGCD-*based scheme*

In this section we describe the construction of homomorphic encryption over the integers, which was originally proposed by Dijk, Gentry, Halevi and Vaikuntanathan [vDGHV10], and was improved many times afterwards [CLT14, CS15a, CNT12]. Among these improvements, we focus on batch computation by extending the original idea to apply the Chinese Remainder Theorem [CCK+13]. The secret key somewhat homomorphic cryptosystem is defined as follows:

Definition 3.1: Let λ be the security parameter and consider the parameters ρ, η, γ as functions of λ. The algorithm KeyGen randomly generates the secret key $\mathtt{dk}$ as an odd integer p with bit-length η. To encrypt a message $m \in \mathbb{Z}_Q$, the algorithm Enc randomly chooses the integers r with ρ bits and q with γ/η bits and computes the ciphertext:

$$c = qp + Qr + m.$$

The decryption algorithm computes the message

$$m = \mathrm{Dec}_{\mathtt{dk}}(c) = [c]_p \pmod{Q}.$$

It is easy to see that the encryption is a ring homomorphism.

Definition 3.2: Consider the following distribution

$$\mathcal{D}_{\gamma,\rho}(p) = \{qp + r \mid \ q \leftarrow \mathbb{Z} \cap [0, 2^\gamma/p), \\ r \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho)\}.$$

Given polynomially many samples from $\mathcal{D}_{\gamma,\rho}(p)$, finding p is a problem called *approximate greatest common divisor (*AGCD*).*

The AGCD problem was studied by Howgrave-Graham in the context of cryptanalysis [HG01]. To obtain a secure cryptosystem, parameters ρ, η, γ must be chosen to resist against attacks described in the literature [Lag85, NS01, HG01, CH11].

Definition 3.3: The symmetric scheme described above can easily be adapted to allow batch operations [KLYC13, CCK+13] as follows. We use notation ℓ for the number of slots in the the plaintext space $\mathcal{M}$ and w for the bit-length of each slot. We assume that (ℓ, w) are fixed and public values. Also, we are going to use secondary security parameters (ρ, η, γ) to describe the cryptosystem. Since these parameters are functions of the primary security parameter λ and are intimately connected to the complexity of the best-known attacks to the AGCD problem, we postpone the concrete description of these functions in order to get a cleaner definition of the scheme.

Key Generation. The $\mathrm{KeyGen}(1^\lambda, f)$ algorithm chooses pairwise coprime integers Q_j with w bits, for $1 \leq j \leq \ell$, and pairwise coprime p_j with η bits, for $0 \leq j \leq \ell$. We have that Q_j represents the size of each plaintext slot, while the plaintext space is given by $\mathcal{M} = \mathbb{Z}_{Q_1} \times \cdots \times \mathbb{Z}_{Q_\ell}$. Note that $\mathcal{M}$ is isomorphic to $\mathbb{Z}_Q$ for $Q = \prod_{j=1}^{\ell} Q_j$. The algorithm computes $p = \prod_{j=0}^{\ell} p_j$. The ciphertext space is given by $\mathcal{C} = \mathbb{Z}_p$. The secret key is given by $\mathtt{dk} = [p_j]$ and the evaluation key is $\mathtt{edk} = p$.

Encryption. Given $[m_j] \in \mathcal{M}$, algorithm $\mathrm{Enc}_{\mathtt{dk}}([m_j])$ chooses a random integer r_0 in the interval

$(-p_0/2, p_0/2]$ and the random integers $r_1, \ldots, r_\ell$ with ρ bits. The ciphertext is computed by

$$c = \text{CRT}(r_0, [m_j + r_j Q_j]),$$

where CRT returns the unique integer modulo p that is congruent to r_0 modulo p_0 and congruent to $m_j + r_j Q_j$ modulo p_j, for every j. Thus the output of the algorithm is equal to $c = \text{ENC}_{\texttt{dk}}(m)$.

Decryption. Given $c \in \mathcal{C}$, the $\text{DEC}_{\texttt{dk}}(c)$ algorithm computes

$$m_j \equiv c \pmod{p_j} \pmod{Q_j},$$

for $1 \leq j \leq \ell$ and outputs $[m_j] = \text{DEC}_{\texttt{dk}}(c)$.

Evaluation. Homomorphic operations are carried out by simply adding and multiplying integers modulo p.

The construction makes use of the following parameters:

- γ is the bit-length of ciphertexts. This parameter must be large enough in order to avoid attacks against AGCD problem, such as the ones derived from Coppersmith's method, as for example Howgrave-Graham [HG01] and Cohn-Heninger [CH11] attacks, the simultaneous Diophantine approximation strategy of Lagarias [Lag85] and Nguyen and Stern's orthogonal lattice [NS01] attack. In summary, we have that these attacks lead to the condition $\gamma = \eta^2 \Omega(\lambda)$, as described in Tancrède Lepoint's PhD thesis [Lep14];
- η is the bit-length of secret key p_j. It must be large enough to accommodate the noise growth after homomorphic operations. However, it must also be quadratic in the security parameter in order to avoid the elliptic curve method (ECM) factorization attack [Len87]. The general number field sieve is another strategy that can be used to factorize the error-free term, but it depends on the size of this term, which is equal to γ. Therefore, since p is the multiplication of many prime numbers, namely ℓ primes p_i, for large ℓ, then γ is too large and it turns out that this strategy is not suitable in this case;
- ρ is the bit-length of the noise r_j. This parameter must be chosen satisfying $\rho = \Omega(\lambda)$, such that the scheme resists attacks against the noise [CN12].

Theorem 1: *The* correctness *of the* AGCD*-based scheme is obtained if η is chosen in order to satisfy the following inequality.*

$$||f||_\infty . 2^{2(\rho+w)} < 2^{\eta-4},$$

where $||f||_\infty$ if given by the infinity norm of the function f that the scheme can homomorphically evaluate.

The proof is given by the straightforward calculation of the noise length after application of the Chinese Remainder Theorem, as detailed in Lemmas 6 and 7 from [KLYC13]. ■

To our purposes, we have that f usually will have additive depth bounded by 2^{20}. Using the Sage routine presented by Lepoint in his PhD thesis [Lep14], we can find parameters that are not vulnerable to the attacks described above. For instance, we have that a possible instantiation of the above parameters to achieve security $\lambda = 80$ would be: $\rho = 96$, $\eta = 351$ and $\gamma \approx 2.67 \times (10)^6$. Using this configuration, the scheme supports at most $\ell = \gamma/(c\eta)$ slots of integers whose bit-length is $w = (\eta - 24)/2 - \rho$, where c is a small constant that is important for security reasons, as we are going to explain in Section 4. With this choice for η, we can set w roughly equal to $\eta/2 - \rho - 12$, because we only need to support one level of multiplication. Hence, as we increase the size of the ciphertext, the plaintext size gets closer and closer to half the size of the ciphertext. Thus, if we define the *overhead* by $\gamma/(w\ell)$, i.e. the ratio between the ciphertext and the plaintext sizes, this overhead tends to 2 as γ increases. However, it would occur only for huge ciphertexts, and in order to obtain practical parameters, it is necessary to choose a feasible ciphertext size, what makes the overhead larger but still interesting when compared to the BGV scheme instantiation.

Theorem 2: *The* AGCD*-based construction described above is secure on the assumption that ACGD problem is hard [CCK+13].*

In the proof of the above theorem, it is crucial that the evaluation key p is made public. The value is an exact multiple of the secrets $[p_j]$ and for this reason is called *noise-free*. With access to this value, it is possible to transform an instance of the AGCD problem to an instance of the *ℓ-decisional* AGCD_Q, proving that one problem can be reduced to the other. It is not known how to do this reduction without knowing the noise-free value and solving this problem would allows us to avoid the requirement of choosing η quadratic in the security parameter, what would lead to much smaller η and consequently the ciphertext size γ could also be chosen to be smaller. Therefore, such a contribution would represent an important improvement for the choice of parameters.

3.3 Homomorphic hash

Homomorphic hashing in the context of verifiable computation is important because it allows to compress the ciphertext produced by the CPA-secure scheme while preserving the ring homomorphism.

Universal one-way homomorphic hash. We say that a function is a universal one-way homomorphic hash if it respects that if κ and κ' are randomly chosen, we have that H is a ring homomorphism and the following property is valid:

for all $c \neq c'$, $\Pr[\mathrm{H}_\kappa(c) = \mathrm{H}_\kappa(c')]$ *is a negligible function of* λ.

Hence, if we want to use homomoprhic encryption where ciphertexts are given by big integers, we can randomly choose κ as an integer of bit-length 2λ and simply reduce the ciphertext modulo κ. It is important to note that the function H_κ leaks information about κ, but this problem is solved by *encoding* H_κ *in the exponent* as we are going to show later.

Definition 3.4: Given the security parameter λ and letting $\mathcal{H}$ be the range of the hash function, whose size is equal to 2λ bits. The keyed family of hash functions $\mathrm{H} : \mathcal{C} \times \mathcal{K} \to \mathcal{H}$ is defined by algorithms $(\text{H.KeyGen}, \mathrm{H}, \text{H.Eval})$ as follows:

Key generation. The algorithm $\text{KeyGen}(1^\lambda)$ outputs a randomly chosen key $\kappa \xleftarrow{\$} \mathcal{K}$ where $\mathcal{K}$ is given by the integers in the interval $[2^{2\lambda-1}, 2^{2\lambda})$.

Hash. Given a ciphertext $c \in \mathcal{C}$, algorithm computes $\mathrm{H}_\kappa(c) = c \pmod{\kappa}$.

Evaluation. Given the description of function f and its inputs $[h_i] \in \mathcal{H}$, for $1 \leq i \leq t$, and considering that f is an algebraic circuit over $\mathcal{H}$, we can describe the H.Eval_κ algorithm by defining addition and multiplication gates over $\mathcal{H}$, which are implemented by the arithmetic modulo κ. Since the map H is a family (indexed by $\mathcal{K}$) of homomorphisms between $\mathcal{C}$ and $\mathcal{H}$, we have that H.Eval_κ correctly computes the output of f over the hashes $[h_i]$.

Theorem 3: *Given the choice of* κ *in the definition of* H_κ *and random elements* $c, c' \in \mathcal{C}$*, we have that* $\Pr[\mathrm{H}_\kappa(c) = \mathrm{H}_\kappa(c')] \leq 1/\lambda$.

The proof is given by the application of the birthday paradox over the range size of the hash function. ■

Definition 3.5: Now we define a *collision-resistant homomorphic hash.* Suppose the existence of an universal one-way homomorphic hash function H_κ with domain equal to the ciphertext space $\mathcal{C}$ of the underlying CPA-secure homomorphic cryptosystem and range equal to $\mathcal{H}$. We show how such function can be used as a black-box to construct a collision-resistant homomorphic hash function that can homomorphically compute many additions and at most one multiplication.

Key generation. Let $\mathbf{bp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e, g, h)$ be a bilinear pairing. The algorithm $\text{KeyGen}(1^\lambda)$ computes $\kappa = \text{H.KeyGen}(1^\lambda)$ and outputs $(\mathbf{bp}, \kappa)$.

Hash. Given the input $c \in \mathcal{C}$, the algorithm $\hat{H}(c)$ outputs

$$(T, U) = (g^{\mathrm{H}_\kappa(c)}, h^{\mathrm{H}_\kappa(c)}).$$

Eval. Given (T_1, U_1) and (T_2, U_2), additions are computed by $(T_1.T_2, U_1.U_2)$ and multiplications are computed by $e(T_1, U_2)$.

Theorem 4: *Let* H *be a universal one-way hash function. Then the construction described above gives us a collision-resistant hash function based on the* Bilinear Diffie-Hellman Inversion *assumption [BBG05], as detailed in Theorem 3 from [FGP14].*

3.4 Amortized closed-form PRF

Employing homomorphic MAC is not enough to achieve our goal. Namely, we want to construct a homomorphic cryptosystem that resists against key recovey attacks and chosen ciphertext attacks. The main reason is that the verification of the MAC requires an algorithm that has complexity roughly the same as computing the function f itself. In [BFR13], new concepts were proposed to achieve *homomorphic MACs with efficient verification.* The construction follows the main ideas proposed in [CF13]. Shortly, given a message $m \in \mathcal{M}$, where $\mathcal{M} = \mathbb{Z}_p$, the idea is to encode it in a degree-1 polynomial $y \in \mathbb{Z}_p[x]$, such that $y(0) = m$ and $y(\alpha) = \mathcal{F}_K(\tau)$, where α is a secret value and $\mathcal{F}_K$ is a PRF function over the label τ, which is used to identify the message. A possible instantiation of the PRF function could be done by using a function F' that in practice can be implemented by the AES symmetric encryption scheme where the key is given by K. Thus, given messages m_i as input of a function f, the corresponding authentication polynomial y_i can be used to compute $y = f(y_1, \ldots, y_t)$, so that y authenticates the computation $m = f(m_1, \ldots, m_t)$. Therefore, it is possible to verify if the computation of f over the authentication tags is correct by verifying if (i) $y(0) = m$ and (ii) $y(\alpha) = f(\mathcal{F}_K(\tau_1), \ldots, \mathcal{F}_K(\tau_t))$. Hence, condition (ii) must be avoided in order to obtain a homomorphic MAC with efficient verification. One could hope to reuse $y(\alpha)$, but as described in the original work [CF13], labels τ_i can not be reused, which turns it impossible to reuse $y(\alpha)$ naively.

To solve the problem described above, Backes et al [BFR13] proposed the concept of *multi-labels*, where an input m_i to the function f is identified by the pair (Δ, τ_i), where Δ identifies the *data set* and τ_i identifies an specific input of f. Thus, τ_i can be reused as soon as we use distinct values for Δ for each computation of f. Actually, this concept is natural in practice. If the messages m_i correspond to the stock market price of a company in a determined moment, and if we want to compute statistical functions of the price after some period of time, we could interpret Δ as a univocal value for each computation of the function f, as the company name together with a period of time and τ_i as the price at different times along this period. Using this idea, it is possible to compute an information y_f that depends only on the input labels τ_i, but not on the data set identification Δ. Such y_f can be used to efficiently verify the authentication tag of any possible data set for f. For the sake of simplicity, we are going to replace notation τ_i by the index i. Thus we have that each argument of the function f is identified by this index i and consequently the corresponding multi-label for a certain computation of f is given by (Δ, i).

Definition 3.6: The PRF family is defined by the algorithms: $(\mathcal{F}.\text{KeyGen}, \mathcal{F})$, where $\mathcal{F}.\text{KeyGen}$ computes a key $K = (K_1, K_2)$ and returns the description of the PRF function $\mathcal{F}_K$, which must satisfy the pseudorandomness property. For instance, for any adversary $\mathcal{A}$, we have that:

$$|\Pr[\mathcal{A}^{\mathcal{F}_K(\cdot)} = 1] - \Pr[\mathcal{A}^{\phi(\cdot)} = 1]|$$

is a negligible function over the security parameter λ and ϕ is a random function.

Also, given the multi-label (Δ, i), the PRF function computes $(u, v) = \mathcal{F}'_{K_1}(i)$ and $(a, b) = \mathcal{F}'_{K_2}(\Delta)$. The algorithm then outputs $R = g^{au+bv}$ and $S = h^{au+bv}$.

Definition 3.7: An *amortized closed-form PRF* is a secure PRF family of functions along with the algorithms: $(\text{FEval}^{\text{OFF}}_{f,i}, \text{FEval}^{\text{ON}}_{f,\Delta})$, such that

- $y_f = \text{FEval}^{\text{OFF}}_{f,i}(K)$ is obtained as follows. For each input i to the function f, compute $(u_i, v_i) = \mathcal{F}_K(i)$ and let $y_i(z_1, z_2) = u_i z_1 + v_i z_2$ be the degree-1 polynomial over the variables z_1 and z_2. Considering that the function f is described by an algebraic circuit, if we replace the gates of this circuit by corresponding gates in the polynomial ring over the variable z_1 and z_2, and if we replace the input m_i of f by the polynomial y_i, we obtain the polynomial $y_f(z_1, z_2)$.
- $W = \text{FEval}^{\text{ON}}_{f,\Delta}$ runs in time $\mathcal{O}(t)$ and works as follows. Compute $(a, b) = \mathcal{F}_K(\Delta)$ and compute $w = y_f(a, b)$ by evaluating the polynomial y_f as previously described, and output $W = e(g, h)^w$.

Definition 3.8: Let $\mathbf{bp}$ be a bilinear map and r_0, r_1, r_2, x_1, x_2 be randomly chosen elements in $\mathbb{F}_q$. Let $T = (g, h, g^{x_1}, g^{x_2}, g^{x_1 r_1}, g^{x_2 r_2}, h^{x_1}, h^{x_2}, h^{x_1 r_1}, h^{x_2 r_2})$.

We say that the decision linear assumption holds if

$$|\Pr[\mathcal{A}(\mathbf{bp}, T, g^{r_1+r_2}, h^{r_1+r_2}) = 1] - \Pr[\mathcal{A}(\mathbf{bp}, T, g^{r_0}, h^{r_0}) = 1]| \quad (1)$$

is negligible in λ.

3.4.1 Homomorphic MAC

Homomorphic MACs were recently formalized by Gennaro and Wich [GW13]. Given a set of messages $[m_i]$, the scheme allows to produce authentication tags $[\sigma_i]$ which can be homomorphically combined in order to authenticate the corresponding combination over the original messages. Such construction must satisfy the requirement of being better than the trivial solution, where after algebraically combining the messages we compute the authentication tag. In the cloud computing scenario, this requirement translates to have lower *communication complexity* than sending the entire set $[m_i]$ to the client. The solution proposed by Gennaro and Wichs are based on fully homomorphic encryption and allows arbitrary computation over the authentication tags. Catalano and Fiore [CF13] proposed a solution that does not offer the same flexibity, but on the other hand achieves better performance. Both schemes makes use of *labeled programs* in order to obtain the security proof. In this section we extend this definition to the multi-label setting.

Definition 3.9: A homomorphic MAC scheme HomMAC is defined as follows:

Key generation. Given the security parameter λ and the description of funtion f, the algorithm $\text{KeyGen}(1^\lambda, f)$ computes both

$$\kappa = \text{H.KeyGen}(1^\lambda) \text{ and } K = \text{PRF.KeyGen}(1^\lambda, f).$$

It also computes $y_f = \text{FEval}^{\text{OFF}}_{f,i}(K)$. The output of the algorithm is $\mathtt{vk} = (\kappa, K, y_f)$.

Authentication. Given the ciphertext c_i corresponding to the multi-label (Δ, i), the algorithm Auth computes

$$(T_i, U_i) = \text{H}_\kappa(c_i) \text{ and } (R_i, S_i) = \text{PRF}_K(\Delta, i),$$

then it computes

$$X_i = (R_i T_i^{-1})^{1/\alpha} \in \mathbb{G}_1, Y_i = (S_i U_i^{-1})^{1/\alpha} \in \mathbb{G}_2$$

and outputs $\delta_i = (T_i, U_i, X_i, Y_i, \Lambda = 1)$.

Evaluation. Given the authentication tags $\sigma_i = (T_i, U_i, X_i, Y_i, \Lambda_i)$, for $i \in \{1, 2\}$, the computation of f over the authentication tags is carried out as follows:

- Additions:

$$T = T_1.T_2,\ U = U_1.U_2,\\ X = X_1.X_2,\ Y = Y_1.Y_2,\\ \Lambda = \Lambda_1.\Lambda_2.$$

- Multiplications:

$$T = e(T_1, U_2), U = e(T_2, U_1), \Lambda = e(X_1, Y_2),$$

$$X = e(X_1, U_2).e(X_2, U_1), Y = e(T_1, Y_2).e(T_2, Y_1).$$

Output $\sigma = (T, U, X, Y, \Lambda)$.

Verification. Given the private key $\mathtt{vk} = (\kappa, K)$, the ciphertext c and its multi-label (Δ, i), run $W = \text{FEval}^{\text{ON}}_{f,\Delta}$ To describe the algorithm Ver we consider two cases:

- if f has degree equal to 2, verify if the following equations hold

$$(T, U) = \text{H}_\kappa(c), X = Y, W = T X^\alpha \Lambda^{\alpha^2};$$

- if f has degree equal to 1, verify if the following equations hold

$$(T, U) = \mathrm{H}(c), e(T, h) = e(g, U),$$

$$e(X, h) = e(g, Y), W = e(TX, h)^{\alpha}.$$

Theorem 5: *Let f be a quadratic multivariate polynomial. Then the homomorphic MAC described above is secure according to definition 2.5.*

Construct the list T as described in definition 2.5. Apply the probabilistic test in Proposition 1 from [CF13] to verify if the function f is well defined with respect to the list T as in **Game 0** from Theorem 3 in [CF13]. Also, Theorem 6 from [FGP14] constructs a sequence of Games that allows to prove that an adversary can not forge the computation of f. Both arguments together are enough to prove that the homomorphic MAC scheme is secure against type-1 and type-2 forges. ■

3.4.2 Verifiable computation

In this section we present the $\mathcal{VC}$ scheme described in [FGP14]. We use the homomorphic MAC defined in last section to organize the description. The advantage of the $\mathcal{VC}$ scheme that we show here is that it is the only private and adaptively secure verifiable computation scheme proposed in the literature, which allows the functionality of verifying computation over encrypted data, but also allows us to transform CPA-secure homomorphic encryption schemes into CVA-secure cryptosystems that can homomorphically evaluate quadratic multivariate polynomials, such that it can be used to outsource computation of statistical functions to the cloud in the highest security model possible.

Definition 3.10: The $\mathcal{VC}$ scheme is defined as follows:

Key generation. The algorithm $\textsc{KeyGen}(1^{\lambda}, f)$ computes

$$(\mathtt{dk}, \mathtt{edk}) = \mathcal{E}_{\mathrm{CPA}}.\textsc{KeyGen}(1^{\lambda}, f),$$

$$\mathtt{vk} = \mathrm{HomMAC}.\textsc{KeyGen}(1^{\lambda}, f)$$

and calculates y_f for the amortized closed-form related to f. Its output is given by $\mathtt{sk} = (\mathtt{dk}, \mathtt{vk}, y_f)$.

Problem generation. Given $[m_i]$ and (Δ, i), such that $m_i \in \mathcal{M}$ and $\mathtt{sk}$, the algorithm $\textsc{ProbGen}$ computes

$$c_i = \mathcal{E}_{\mathrm{CPA}}.\textsc{Enc}_{\mathtt{dk}}(m_i) \text{ and}$$

$$\sigma_i = \mathrm{HomMAC}.\textsc{Auth}_{\mathtt{vk}}(c_i, \Delta, i).$$

Evaluation. Given the evaluation key $\mathtt{esk} = (\mathtt{edk}, \mathtt{evk})$, the ciphertexts $[c_i]$ and the authentications tags $[\sigma_i]$ and the description of function f, algorithm $\textsc{Compute}$ uses the homomorphic property of $\mathcal{E}_{\mathrm{CPA}}$ and HomMAC to evaluate the function f. Namely, it computes

$$c = \mathcal{E}_{\mathrm{CPA}}.\textsc{Eval}_{\mathtt{edk}}([c_i], f),$$

$$\sigma = \mathrm{HomMAC}.\textsc{Eval}_{\mathtt{evk}}([\sigma_i], f)$$

and outputs (c, σ).

Verification. Given $\mathtt{sk} = (\mathtt{dk}, \mathtt{vk})$, y_f and Δ, the algorithm $\textsc{Verify}$ computes $W = \mathrm{FEval}^{\mathrm{ON}}_{f, y_f, \Delta}$ and returns

$$\mathrm{HomMAC}.\textsc{Ver}_{\mathtt{vk}}(W, c, \sigma).$$

Theorem 6: *If $\hat{H}$ is a collision resistant hash function and $\mathcal{E}_{\mathrm{CPA}}$ is a CPA-secure homomorphic encryption scheme, the $\mathcal{VC}$ scheme described above is correct, adaptatively secure and input private. The proof is detailed in Theorem 6 from [FGP14].*

3.5 The secret key encryption scheme

The secret key encryption scheme that can homomorphically compute the quadratic multivariate polynomial f over encrypted input is defined as follows:

Key generation. Given the description of the function f, let $\mathcal{E}_{\mathrm{CPA}}$ be a CPA-secure secret key homomorphic encryption scheme and let $\mathcal{VC}$ be a private and adaptively secure verifiable computation scheme as previously defined and containing the inherent message authentication algorithm HomMAC. We compute $(\mathtt{dk}) = \mathcal{E}_{\mathrm{CPA}}.\textsc{KeyGen}(1^{\lambda}, f)$ and $(\mathtt{sk}, \mathtt{esk}) = \mathcal{VC}.\textsc{KeyGen}(1^{\lambda}, f)$. The secret key is given by $(\mathtt{dk}, \mathtt{sk})$ and the evaluation key is given by $(\mathtt{edk}, \mathtt{esk})$.

Encryption. For $m \in \mathcal{M}$ and multi-label (Δ, i), if the multi-label was not previously used, compute $c = \mathcal{E}_{\mathrm{CPA}}.\textsc{Enc}_{\mathtt{dk}}(m)$, compute

$$\sigma = \mathrm{HomMAC}.\textsc{Auth}_{\mathtt{vk}}(c, \Delta, i)$$

and output $(c, \sigma, \Delta, i, f_{\mathrm{ID}})$.

Decryption. For $(c, \sigma, \Delta, i, f^{\star}) \in \mathcal{C}$, when $f^{\star}$ is the identity function f_{ID}, if $\mathrm{HomMAC}.\textsc{Ver}_{\mathtt{vk}}(\sigma, c, \Delta, i)$ rejects then return $\perp$. When $f^{\star} = f$, if $\mathcal{VC}.\textsc{Verify}_{\mathtt{sk}}(\sigma, \Delta, f)$ rejects then return $\perp$, otherwise return

$$m = \mathcal{E}_{\mathrm{CPA}}.\textsc{Dec}_{\mathtt{dk}}(c).$$

Evaluation. Given the evaluation key $\mathtt{esk}$ and $([c_i], [\sigma_i], f)$, for $1 \leq i \leq t$, return (c, σ), where

$$(c, \sigma) = \mathcal{VC}.\textsc{Compute}_{\mathtt{esk}}([c_i], [\sigma_i], f).$$

3.6 Security proof

Theorem 7: *The scheme described above is CVA-secure based on the assumption that $\mathcal{E}_{\mathrm{CPA}}$ is CPA-secure and $\mathcal{VC}$ is a private and adaptively secure verifiable computation scheme.*

The proof for Theorem 1 in the FGP scheme [FGP14] considers an adversary that can break the CPA-security of the underlying homomorphic encryption if he can obtain information about the input of the $\mathcal{VC}$ scheme. In order to do that, a simulator is constructed, such that it has access to two oracles that answer (i) encryption queries and (ii) verification queries. Both

oracles are sufficient to prove security in the context of verifiable computation, but regarding the chosen ciphertext security model, it is necessary to show how to simulate the decryption oracle also. Hence, if we want to construct an encryption scheme based on verifiable computation, we must provide a way to simulate the decryption oracle and then we can use the hybrid argument described in Theorem 1 of the FGP scheme. Shortly, the adversary can use the oracles to distinguish between two ciphertexts that differ only in one position and the hybrid argument is used to extend this distinguishing algorithm for the general case. The scheme $\mathcal{E}_{\text{CPA}}$ is secret key and the simulator can use the encryption oracle to obtain ciphertexts during the experiment. Moreover, only the simulator can authenticate these ciphertexts, since he is the owner of the $\mathcal{VC}$ key pair. On the other hand, while the $\mathcal{E}_{\text{CPA}}$ secret key can be used to decrypt ciphertexts, the simulator can recompute the function f using the original messages, since it can first verify if the decryption query is a valid computation or a ciphertext obtained from the encryption oracle. Since an adversary cannot authenticate ciphertexts, the decryption oracle is useless to him. The simulator can recompute the function f for each decryption query during the experiment.

Suppose there is an adversary $\mathcal{A}$ that wins the CVA game with non-negligible advantage. We will show that the simulator can construct an adversary $\mathcal{A}^\star$ that wins a CPA game with non-negligible advantage. The simulator computes $\mathtt{sk} = \mathcal{VC}.\textsc{KeyGen}(1^\lambda, f)$. When $\mathcal{A}$ asks the encryption of a message m, the simulator forwards it to the encryption oracle of the CPA experiment, which returns c. The simulator then computes $\sigma = \text{HomMAC}.\textsc{Auth}_{\mathtt{vk}}(c, \Delta, i)$ and returns $(c, \sigma, \Delta, i, f_{\text{ID}})$ to $\mathcal{A}$. The simulator also stores the tuple $(c, \sigma, \Delta, i, m)$ for future usage.

To simulate the decryption oracle under query $(c, \sigma, \Delta, i, f^\star)$, when $f^\star = f_{\text{ID}}$, if the multi-label (Δ, i) was previously queried to the encryption oracle, the simulator returns the original message m, otherwise it returns $\perp$. Since the probability that $\text{HomMAC}.\textsc{Ver}_{\mathtt{sk}}(c, \Delta, i, \sigma) \neq \perp$ is negligible, the event of an adversary generating a valid authentication tag for a fresh ciphertext c will not help him to win the game. When $f^\star = f$, if $\mathcal{VC}.\textsc{Verify}_{\mathtt{sk}}(\sigma, \Delta) = \perp$ the simulator returns $\perp$ to the adversary. Otherwise it returns the evaluation of f over the corresponding plaintexts, which were previously stored. Since HomMAC is unforgeable and $\mathcal{VC}$ is private and adaptively secure, the probability that $\mathcal{A}$ queries the decryption oracle using a ciphertext that was not returned by the encryption oracle and is the result of a forged homomorphic computation of function f is negligible. Furthermore, the probability that an adversary submits a query which is the result of a computation that is not the one defined by the function f, such that this submission is valid according the verifiable computation scheme, is also negligible. Therefore the simulation of the decryption oracle is indistinguishable from the real decryption algorithm. Finally, $\mathcal{A}$ generates the challenge messages m_0 and m_1, which the simulator forwards to the CPA scheme, that outputs the challenge ciphertext $c^\star$. The simulator forwards the challenge to $\mathcal{A}$. If $\mathcal{A}$ has non-negligible advantage to win the CVA game, then the simulator uses $\mathcal{A}$ output to construct $\mathcal{A}^\star$ that wins the CPA game also with non-negligible advantage. ■

Multi-labels (Δ, i) are important in the proof of Theorem 6 in the original paper [FGP14], when the general black-box scheme described in the original paper [FGP14] is instantiated for quadratic multivariate polynomials, because the authentication tags are homomorphic only for the same Δ, that represents an specific input to the function f. Also, since multi-labels cannot be reused, an attacker cannot compute the PRF for a new multi-label, because it requires the private key of the PRF function. Therefore the only way to use previously queried ciphertexts, with its underlying multi-labels, is by honestly computing the function f.

4 Choice of parameters

We followed the ideas presented in Tancrède Lepoint PhD thesis [Lep14] in order to choose parameters to our scheme. Firstly, we compute η to avoid the ECM factorization attack. Then we establish an upper bound to ρ and γ. Thus we can compute the ciphertext size to resist against the orthogonal lattice attack by decreasing the value of the noise parameter ρ until it is secure against Chen and Nguyen attack. Afterwards, since we have calculated the values of ρ, η and γ as in Table 1, we can obtain the scheme parameters ℓ and w, with which we can calculate the plaintext size and the overhead of the scheme. We implemented a routine in Sage to calculate these values and the result is shown in Table 2. To compute ℓ we had to decrease the size of the ciphertext from γ to $\gamma = \gamma' - (\ell - 1)\eta$. In [CCK+13], the authors show that the $\text{AGCD}_{1,\gamma}$ problem, the usual approximate GCD problem, can be reduced to the $\text{AGCD}_{\ell,\gamma'}$ problem, i.e. the CRT-based construction defined in Section 3.2. Then we have recomputed the ciphertext size using the relation $\gamma' = 1.5\gamma$ and calculated the number of slots using $\ell = \gamma/2\eta$.

λ	ρ	η	γ (MBits)
80	96	351	1.78
112	94	475	3.27
128	92	603	5.28

Table 1 AGCD parameters

If we want to obtain smaller ciphertext, then we can use the relations $\gamma' = 1.1\gamma$ and $\ell = \gamma/10\eta$, what results in a larger overhead, as shown in Table 3.

This result shows that actually the AGCD-based scheme is competitive with BGV. Table 4 presents a comparison between both proposals and although the ciphertext in first choice is bigger than in the BGV-based

λ	γ' (MBits)	ℓ	w	m (KBits)	overhead
80	2.67	2535	67	170	15.70
112	4.90	3442	131	451	10.87
128	7.92	4378	197	862	9.18

Table 2 Low overhead configuration

λ	γ' (MBits)	ℓ	w	m (KBits)	overhead
80	1.95	507	67	34	57.40
112	3.59	688	131	90	39.88
128	5.80	875	197	172	33.72

Table 3 Smaller ciphertext configuration

construction, it has better overhead. On the other hand, we can use a second choice of parameters, such that the overhead is worse, but the ciphertext size is roughly the size of the BGV-based scheme.

λ	scheme	overhead	γ (MBits)	m (KBits)
80	first choice	15.70	2.67	170.0
80	second choice	57.40	1.95	34.0
80	[FGP14]	346.15	1.80	5.2
128	first choice	9.18	7.92	862.0
128	second choice	33.72	5.80	172.0
128	[FGP14]	545.45	4.80	8.8

Table 4 Comparison for $\lambda = 80$ and $\lambda = 128$

By measuring the overhead of the homomorphic encryption scheme using the ratio between the ciphertext and plaintext sizes, we are assuming that integer multiplication is as efficient as polynomial ring multiplication. Although such assumption is not true, the difference between the complexity of the multiplication algorithm for these two schemes is only polylogarithmic. Namely, big integers can be multiplied in complexity $O(n \log n \log\log n)$ using the Schönhage–Strassen algorithm. On the other hand, after using the FFT algorithm, BGV-based ciphertexts can be multiplied basically in linear time. Thus, our proposal offer a tradeoff between the ciphertext size, which imply worse performance for arithmetic operations and the overhead given by the amount of information that can be encoded inside each ciphertext. Since the verifiable computation scheme depends on bilinear pairings to generate and verify authentication tags, the fact that we can encode more information inside the ciphertext is important because computing pairings is computationally expensive and then a better overhead means that we need less bilinear pairings to verify that the computation was carried out as we wanted.

Our proposal has small overhead, since it is possible to encode a lot of information inside each ciphertext. The analysis allows to conclude that the ability to encode more information inside each ciphertext is indeed important, because for a small amount of bilinear pairings we can verify the computation of many slots in parallel. Nevertheless, the number of slots is a parameter that depends strongly on the target application. Hence, if a high number of slots is not necessary, then we can use the strategy described above to choose an appropriate value for ℓ, such that the ciphertext is minimal.

5 Related work

Homomorphic encryption over quadratic multivariate polynomials was already possible via the BGN scheme [BGN05]. The proposal is semantically secure and, although not proved to be CCA1-secure, it is not known to be vulnerable to CCA1 attacks or key recovery attacks. Proving that the BGN scheme is indeed CCA1-secure or finding a CCA1 attack is an interesting open problem.

The BGN construction employs bilinear pairings to allow one multiplication of ciphertexts and, since it solves the same problem we are targeting, it is important to consider the computational cost of this solution when compared to the verifiable computation scheme. A restriction of this cryptosystem is that it must have a small plaintext size, because the final step in the decryption algorithm is to compute a discrete logarithm. A reasonable plaintext space for the BGN scheme is $\{0,1\}^{10}$. On the other hand, the verifiable computation scheme works with a much larger plaintext space. For instance, at the 80-bit security level, Fiore et al [FGP14] suggest parameters for computation over 165 slots of 32 bits. Thus, it would be necessary to use at least 3 BGN ciphertexts for each slot and, considering the schoolbook multiplication method, we would need at least 9 bilinear pairings per slot to homomorphically compute a multiplication of two 32-bit messages. Finally, the 165 slots would require roughly 1500 pairing computations, while the verifiable computation scheme uses only a constant number of pairings. Namely, it requires 6 pairings to do the homomorphic multiplication and 3 pairings for the verification algorithm.

The main problem of the BGN construction is that it can deal only with small plaintext size. A solution to this problem is using the Chinese Remainder Theorem to encode more information inside each ciphertext. By making use of public composite moduli, Eom et al [SKEL16] described how to achieve this feature, allowing to encode 9 slots inside the BGN encrypted message. This parallel computation permits to reduce the 1500 pairings computation mentioned in last paragraph to 167 pairings, and maybe now it would be better to use this scheme instead of the $\mathcal{VC}$ scheme. Firstly, to answer this question, we must know how large is the cost of the pairing computation. This question is not simple to answer, because it depends on many parameters, as for example the security level, the type of the pairing and other characteristics. To obtain the same performance of the $\mathcal{VC}$ scheme, the pairing comutation

should cost roughly 1/3 ms, what is reasonable if we use an state-of-art implementation [ABLR13]. But in order to implement Eom et al construction, the pairing computation is not so efficient. In 2013, Aurore Guillevic [Gui13] implemented the BGN scheme both in the composite modulus setting and in the prime order setting, showing that the later is considerable faster than the former, that leads to an implementation that takes 3364 ms to homomorphically multiply two ciphertexts using a variation of the BGN scheme. Therefore, such a scheme could be used if the target application requires a small number of slots.

Another related work is the proposal presented by Gentry, Halevi and Vaikuntanathan [GHV10] that constructs a BGN-type scheme which security is based on the LWE problem. As in the previous case, the scheme is not proven to be CCA1-secure, but it is not known if this scheme is vulnerable to CCA1 or key recovery attacks. The ciphertexts are given by matrices and, in order to follow the strategy of the FGP scheme, it would be necessary to provide an universal one-way homomorphic hash function mapping matrices to integers modulo a random value. However, since there is no such a ring homomorphism between the two underlying mathematical structures, then it is not straightforward to use this scheme together with the verifiable computation construction. Moreover, the ciphertext is a matrix of dimension $m = 8n \log q$ over $\mathbb{Z}_q$, where n and q must be chosen to make the LWE problem hard and also to allow enough number of homomorphic operations. Therefore, since the plaintext is $\mathbb{Z}_2^{m \times m}$, the ratio between ciphertext and plaintext size is given by q, what is much bigger than what can be achieved using the BGV scheme as described in the FGP construction. Also, in order to encode integers or polynomials into ciphertexts we have a considerable waste of plaintext space. Finally, we have that the dimension m is too big and this issues make this scheme not suitable to be used in the context we are interested in.

6 Applications

The scheme described in this paper can be used to compute quadratic multivariate polynomials. One important application is the calculation of statistical functions, as for example average, variance, covariance, standard deviation and linear regression. This statistical functions could be used for instance in the context of healthcare data, because it is possible to compute these functions over encrypted data. Another interesting scenario is the computation of this statistics in the stock exchange context, as described in the work of Backes, Fiore, and Reischuk [BFR13].

In the FGP paper [FGP14] the authors presented ad-hoc constructions to solve different problems as for example is case of computing polynomials of large degree in one variable and linear functions over certain algebraic structures. Such kind of computation can be used to measure distances and correlations on encrypted data. It can also be used to compute the discrete Fourier transform on encrypted data and these applications are obtained by using different assumptions that were not considered in the scope of this work, but they are interesting examples of efficient constructions of verifiable computation.

7 Conclusions

This work presented the detailed description of homomorphic cryptographic primitives that can be plugged in together to construct a CCA1-secure secret key homomorphic encryption scheme, which can be used to delegate computation to a cloud computing service. Homomorphic encryption over the integers was used to achieve a better overhead to build verifiable computation. We also obtained a relatively small ciphertext size and big plaintext space when compared to the FGP proposal for the same security level.

References

[ABLR13] Diego F. Aranha, Paulo S. L. M. Barreto, Patrick Longa, and Jefferson E. Ricardini. The realm of the pairings. In *Selected Areas in Cryptography*, volume 8282 of *Lecture Notes in Computer Science*, pages 3–25. Springer, 2013.

[AG] D. F. Aranha and C. P. L. Gouvêa. RELIC is an Efficient Library for Cryptography. `http://code.google.com/p/relic-toolkit/`.

[BBG05] D. Boneh, X. Boyen, and E. Goh. Hierarchical identity based encryption with constant size ciphertext. In R. Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer Berlin Heidelberg, 2005.

[BFR13] M. Backes, D. Fiore, and R. M. Reischuk. Verifiable delegation of computation on outsourced data. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, CCS '13, pages 863–874, New York, NY, USA, 2013. ACM.

[BGN05] D. Boneh, E. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In *Proceedings of the Second International Conference on Theory of Cryptography*, TCC'05, pages 325–341, Berlin, Heidelberg, 2005. Springer-Verlag.

[BGV12] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully

homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 309–325, New York, NY, USA, 2012. ACM.

[CCK+13] J. Cheon, J. Coron, J. Kim, M. Lee, T. Lepoint, M. Tibouchi, and A. Yun. Batch fully homomorphic encryption over the integers. In T. Johansson and P. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 315–335. Springer Berlin Heidelberg, 2013.

[CF13] D. Catalano and D. Fiore. Practical homomorphic MACs for arithmetic circuits. In T. Johansson and P. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 336–352. Springer Berlin Heidelberg, 2013.

[CFGN15] D. Catalano, D. Fiore, R. Gennaro, and L. Nizzardo. Generalizing homomorphic MACs for arithmetic circuits. Cryptology ePrint Archive, Report 2015/396, 2015. `http://eprint.iacr.org/`.

[CH11] H. Cohn and N. Heninger. Approximate common divisors via lattices. Cryptology ePrint Archive, Report 2011/437, 2011. `http://eprint.iacr.org/`.

[CKV10] K. Chung, Y. Kalai, and S. Vadhan. Improved delegation of computation using fully homomorphic encryption. In T. Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 483–501. Springer Berlin Heidelberg, 2010.

[CLT14] J. Coron, T. Lepoint, and M. Tibouchi. Scale-invariant fully homomorphic encryption over the integers. In H. Krawczyk, editor, *Public-Key Cryptography – PKC 2014*, volume 8383 of *Lecture Notes in Computer Science*, pages 311–328. Springer Berlin Heidelberg, 2014.

[CN12] Y. Chen and P. Nguyen. Faster algorithms for approximate common divisors: Breaking fully-homomorphic-encryption challenges over the integers. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 502–519. Springer, 2012.

[CNT12] J. Coron, D. Naccache, and M. Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 446–464. Springer Berlin Heidelberg, 2012.

[CS15a] J. Cheon and D. Stehlé. Fully homomophic encryption over the integers revisited. In E. Oswald and Marc F., editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Computer Science*, pages 513–536. Springer Berlin Heidelberg, 2015.

[CS15b] A. Costache and N. P. Smart. Which ring based somewhat homomorphic encryption scheme is best? Cryptology ePrint Archive, Report 2015/889, 2015. `http://eprint.iacr.org/`.

[DGM15] R. Dahab, S. Galbraith, and E. Morais. Adaptive key recovery attacks on NTRU-based somewhat homomorphic encryption schemes. In A. Lehmann and S. Wolf, editors, *Information Theoretic Security*, volume 9063 of *Lecture Notes in Computer Science*, pages 283–296. Springer International Publishing, 2015.

[FGP14] D. Fiore, R. Gennaro, and V. Pastro. Efficiently verifiable computation on encrypted data. Cryptology ePrint Archive, Report 2014/202, 2014. `http://eprint.iacr.org/`.

[Gen09] C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`.

[GGP10] R. Gennaro, C. Gentry, and B. Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In T. Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 465–482. Springer Berlin Heidelberg, 2010.

[GHV10] C. Gentry, S. Halevi, and V. Vaikuntanathan. A simple BGN-type cryptosystem from LWE. In H. Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 506–522. Springer Berlin Heidelberg, 2010.

[GPS08] S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113 – 3121, 2008. Applications of Algebra to Cryptography.

[Gui13] A. Guillevic. Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In M. Jacobson, M. Locasto, P. Mohassel, and R. Safavi-Naini, editors, *Applied Cryptography and Network Security: 11th International Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings*, pages 357–372, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[GW13] R. Gennaro and D. Wichs. Fully homomorphic message authenticators. In K. Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013: 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II*, pages 301–320, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[HG01] N. Howgrave-Graham. Approximate integer common divisors. In *CaLC*, pages 51–66, 2001.

[KL07] J. Katz and Y. Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

[KL15] M. Kim and K. Lauter. Private genome analysis through homomorphic encryption. *Journal of Medical Informatics and Decision Making*, 2015.

[KLYC13] J. Kim, M. S. Lee, A. Y., and J. Cheon. CRT-based fully homomorphic encryption over the integers. Cryptology ePrint Archive, Report 2013/057, 2013. `http://eprint.iacr.org/`.

[KU16] Mehmet Sabır Kiraz and Osmanbey Uzunkol. Still wrong use of pairings in cryptography. Cryptology ePrint Archive, Report 2016/223, 2016. `http://eprint.iacr.org/`.

[Lag85] J. C. Lagarias. The computational complexity of simultaneous diophantine approximation problems. *SIAM J. Comput.*, 14(1):196–209, February 1985.

[Len87] H. W. Lenstra. Factoring integers with elliptic curves. *The Annals of Mathematics*, 126(3):649–673, November 1987.

[Lep14] T. Lepoint. *Design and Implementation of Lattice-Based Cryptography*. PhD thesis, École Normale Supérieure and University of Luxembourg, June 2014.

[LMSV12] J. Loftus, A. May, N. P. Smart, and F. Vercauteren. On CCA-secure somewhat homomorphic encryption. In A. Miri and S. Vaudenay, editors, *Selected Areas in Cryptography*, volume 7118 of *Lecture Notes in Computer Science*, pages 55–72. Springer Berlin Heidelberg, 2012.

[NLV11] M. Naehrig, K. Lauter, and V. Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, CCSW '11, pages 113–124, New York, NY, USA, 2011. ACM.

[NS01] P. Nguyen and J. Stern. The two faces of lattices in cryptology. In J. H. Silverman, editor, *Cryptography and Lattices: International Conference, CaLC 2001 Providence, RI, USA, March 29–30, 2001 Revised Papers*, pages 146–180, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[RAD78] R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation, Academia Press*, pages 169–179, 1978.

[Rot10] R. Rothblum. Homomorphic encryption: from private-key to public-key. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:146, 2010.

[SKEL16] H.-S. Lee S. K. Eom and S. Lim. Message expansion of homomorphic encryption using product pairing. *ETRI Journal*, 38(1):123–132, Feb 2016.

[vDGHV10] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *Proceedings of the 29th Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'10, pages 24–43, Berlin, Heidelberg, 2010. Springer-Verlag.

Chapter 3

Discussion

> Science is built up of facts, as a house is built of stones; but an accumulation of facts is no more a science than a heap of stones is a house.
>
> Henri Poincaré

In this chapter we analyze concrete instantiations of AGCD-based and LWE-based schemes, using the best-attack running time to estimate the number of operations that are necessary to break the cryptosystems. Hence, using this method we can understand how to choose parameters to construct practical cryptographic primitives.

3.1 Security of the AGCD problem

The AGCD problem consists in finding a common divisor to a set of approximate multiples of this divisor. Supposing the existence of an algorithm to compute one bit from the plaintext, given just the ciphertext and public parameters, we can use the Binary GCD algorithm to construct a solution to the AGCD problem. The details of the security proof for AGCD-based schemes can be found in paper [96], where the authors show that the existence of an attack to the proposed cryptosystem leads to a solution to the AGCD problem.

In order to define the AGCD problem we consider the following distribution.

$$\mathcal{D}_{\gamma,\rho}(p) = \{pq + r \mid q \leftarrow \mathbb{Z} \cap [0, 2^{\gamma}/p), r \leftarrow \mathbb{Z} \cap (-2^{\rho}, 2^{\rho})\}.$$

Definition 3.1.1. Given parameters (ρ, η, γ) and a polynomial number of elements from the distribution $\mathcal{D}_{\gamma,\rho}(p)$, for a randomly chosen odd integer p, the ***AGCD problem***, consists in revealing p.

```
Lagarias := function(X, X0, rho)
    Z = IntegerRing();
    t := #X;
    XX := Matrix( Z, t, 1, [x : x in X] );
    M := ZeroMatrix( Z, t, t );
    M[1,1] := rho;
    for i:=1 to t-1 do
        M[1,i+1] := XX[i,1];
        M[i+1,i+1] := -X0;
    end for;
    L := LLL(M);
    q0 := (EuclideanNorm(L[1,1]))/(rho);
    if (q0 ne 0) then
        r0 := (X0 mod q0);
        p := (X0-r0) div q0;
        return p;
    end if;
    return -1;
end function;
```

Table 3.1: Lagarias attack

The first strategy one could imagine to solve the AGCD problem is using brute force to find the noise of an arbitrary pair of samples. Considering that the noise has ρ bits, we must find two arbitrary noises r_1 and r_2 from samples $x_1 = q_1 p + r_1$ and $x_2 = q_2 p + r_2$, respectively, compute the GCD of $x_1 - r_1$ and $x_2 - r_2$ and verify if the result has η bits. The complexity of this algorithm is $2^{2\rho}$.

Another immediate attack would be factoring noise-free term $x_0 = pq$ to find p. The best algorithm available to solve this problem is Lenstra's factoring algorithm [60], which has complexity $O(\sqrt{\eta})$. Therefore, choosing $\eta = \lambda^2$ we avoid this kind of attack.

Lagarias attacks. The AGCD problem can be interpreted as the ***simultaneous diophantine equations*** problem and we can use Lagarias' algorithm to solve it. Shortly, the idea of the attack is very simple. We must use samples contained in the public key, namely x_i, for $0 \leq i \leq t$, to compute the lattice given by the following matrix:

$$M = \begin{pmatrix} 2^\rho & x_1 & x_2 & \dots & x_t \\ & -x_0 & & & \\ & & -x_0 & & \\ & & & -x_0 & \\ & & & & -x_0 \end{pmatrix} \tag{3.1}$$

Then, we reduce the lattice basis using, for example, the LLL algorithm as described in the Magma code in Table 3.1. If the reduction is good enough we get a vector $v = \langle q_0 2^\rho, q_0 x_1 - q_1 x_0, \dots \rangle$. Thus we compute the quotient q_0, from which it is possible to obtain the private key p even in the case that x_0 has a non-zero noise r_0.

Other two attacks are important to be considered. For instance, the ***Cohn-Heninger attack*** and the ***orthogonal lattice attack***. Both are described in detail in Lepoint's PhD thesis [62], where a Sage routine is presented, such that it is possible to derive parameters using estimations for the number of operations required to run each attack and thus obtaining the security level of the cryptosystem by making these attacks run in at least 2^{λ} operations.

3.2 Security of the LWE problem

In 2011, Brakerski and Vaikuntanathan [23] proposed two new ideas to help the noise management for homomorphic schemes: ***dimension reduction*** (also known as ***relinearization*** or ***key switching***) and ***modulus reduction***. These new techniques provided a better alternative than the one used to squash the decryption circuit, namely the utilization of the SSSP problem.

These concepts are fundamental to the construction proposed by Brakerski, Gentry and Vaikuntanathan, called ***BGV*** [20], which is a construction with better performance in practice, as we are going to show. Previous constructions have worse methods to deal with multiplications, because the noise grows too fast. Indeed, BGV's noise management avoids the exponential growth inherent in other proposals.

The security of LWE-based cryptosystems follows originally from a quantum reduction to GAP$\mathbf{SVP}_{\gamma}$ in the worst case [87], where γ is a polynomial function on the LWE parameters. A classical reduction was shown by Lindner and Peikert [64], but imposing a condition on the size of the modulus, namely an exponential modulus, which was shown to be not necessary by Brakerski et al in 2013 [22].

Next, we define the LWE problem and present the BGV scheme, which can be easily implemented using a library such as NTL [2], for number theoretic calculations, over GMP [1], for efficient arbitrary precision arithmetic.

3.2.1 The LWE problem

Definition 3.2.1. The ***LWE problem*** consists in finding the vector $s \in \mathbb{Z}_q^n$, given the equations

$$\begin{array}{rcl} \langle s, a_1 \rangle & \approx_{\mathcal{D}} & b_1 \pmod q \\ \langle s, a_2 \rangle & \approx_{\mathcal{D}} & b_2 \pmod q \\ & \vdots & \end{array}$$

Notation $\approx_{\mathcal{D}}$ means a tolerance in the equality, according to a distribution $\mathcal{D}_{n,\sigma,q}$. Namely, $\langle s, a_i \rangle$ has a distance to b_i and this distance is determined by the distribution $\mathcal{D}_{n,\sigma,q}$, generally an n-dimensional Gaussian distribution with standard deviation given by σ. Alternatively, we can write $\langle s, a_i \rangle = b_i + e_i \pmod q$, where $e_i \in \mathcal{D}_{n,\sigma,q}$.

Gaussian distribution plays a central role, because Micciancio and Regev [75] presented in 2004 a concept called ***smoothing parameter***. This parameter allows for a different way for obtaining pseudorandomness from lattices. Figure 3.2.1 shows a simplification of the idea. It shows centered Gaussians, with increasing standard deviation, reduced modulo the trivial one-dimensional lattice $\mathcal{L} = \mathbb{Z}$.

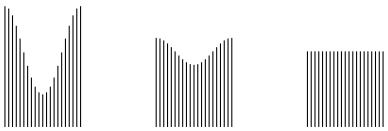

Figure 3.2.1: Gaussian distributions modulo 1

The number of equations do not contribute a lot to the solution. There is a trade-off between the number of equations and the running time to find a solution to the problem; even with an arbitrary number of equations the complexity is at least subexponential [13].

In 2005, Regev [88] presented a quantum reduction from the LWE problem to worst case problems over lattices. Moreover, this work provided what was necessary to construct a new cryptosystem, whose performance is considerably better than other schemes based on lattices. Lindner and Peikert [64] showed a classical reduction, proposing the parameter analysis that is adopted here.

Lyubaskevsky, Peikert and Regev defined a similar version of the LWE problem, but using polynomial rings [68]. This construction is called ***ring LWE***. Concretely, let $f(x) = x^n + 1$, where n is a power of 2. Given an integer q and an element $s \in R = \mathbb{Z}_q[x]/f(x)$, the **ring LWE problem** over R, with respect to the distribution $\mathcal{D}_{n,\sigma,q}$, is defined similarly, that is, it is necessary to find s that satisfies the following equations:

$$\begin{aligned} s.a_1 &\approx_{\mathcal{D}} b_1 \pmod{R_q} \\ s.a_2 &\approx_{\mathcal{D}} b_2 \pmod{R_q} \\ &\vdots \end{aligned}$$

where a_i and b_i are elements of the ring R_q. Modular reduction in R_q is the same as reducing modulo $f(x)$ and reducing the coefficients modulo q.

3.2.2 Somewhat homomorphic encryption

Let $\mathcal{D}_{n,\sigma,q}$ be a n-dimensional spherical discrete Gaussian distribution [68,69] with standard deviation σ and over the ring $\mathbb{Z}_q$.

Definition 3.2.2. **G**iven the security parameter λ and a secondary parameter μ, we choose an integer q with μ bits and $N = \lceil 3 \log q \rceil$. The scheme $\mathcal{E}$ parameters are given by $\lambda, \mu, n, \sigma, q$ and the ring $R = \mathbb{Z}_q[x]/f(x)$ and $f(x)$ a cyclotomic polynomial.

Key Generation. Use the distribution $\mathcal{D}_{n,\sigma,q}$ to compute the polynomial $s^\star \in R$. Denote by s the vector formed by the polynomials 1 and $s^\star$. The private key is given by $sk = s$. Generate randomly a matrix A' with N rows and one column, whose elements are polynomials with coefficients uniformly chosen in $\mathbb{Z}_q$. Use the distribution $\mathcal{D}_{n,\sigma,q}$ to generate N polynomials e_i and compute $b = A's^\star + te$. Compute the matrix A of two columns, the first one equal to b and the second one equal to $-A'$. The public key is given by $pk = A$. By construction, we have that $As = te$.

Encryption. Given a message $m \in \{0,1\}^t$ and the public key pk, we define the matrix m' with two rows, where the first one is the value m and the second one is equal to zero. Generate randomly a column matrix denoted by r with N rows composed by binary polynomials. Finally, output the ciphertext

$$c = \text{ENC}_{pk}(m) = m' + A^T r \pmod{q}.$$

Decryption. Compute

$$m = \text{DEC}_{sk}(c) = [[\langle c, sk \rangle]_q]_t.$$

The correctness of this scheme is easily verified using the relation $As = te$ and the fact that q is chosen sufficiently big such that the error do not get above $q/4$, similarly to the case over the integers. Also, ciphertext addition is done component-wise, while ciphertext multiplication is done with the tensor technique, which will be explained in Section 3.2.6.

```
lwe_attack := function(n,m,q,A,b)
        qI := ZeroMatrix(Z, m, m);
        for i := 1 to m do qI[i,i] := q; end for;
        A1 := VerticalJoin(Transpose(A),qI);
        X, U := HermiteForm(A1);
        B := RowSubmatrixRange(X,1,m);
        BB := LLL(B: TimeLimit:=1);
        L := Lattice(BB);
        tb := Transpose(b)[1];
        w := ClosestVectors(L,tb : TimeLimit:=1);
        e := Matrix(Z,m,1,[b[i,1]-w[1,i] : i in [1..m]]);
        return e;
end function;
```

Table 3.2: LWE attack

3.2.3 LWE security

Gentry, Halevi and Smart [47] used the BGV scheme to homomorphically evaluate the AES symmetric encryption system. They based the security analysis on Lindner and Peikert's [64] work, showing that an LWE-based cryptosystem is secure as long as

$$d > \log\left(\frac{q}{\sigma}\right)\frac{\lambda + 110}{7.2}. \tag{3.2}$$

When applied to homomorphic schemes, this relation acquires a challenging aspect, because as the standard deviation increases, less homomorphic operations can be evaluated, since a larger initial noise would be rapidly propagated, what would require a larger modulus q, depending on the circuit's multiplicative depth. Thus, as the ratio q/σ determines the LWE-based cryptography security, in order to avoid managing the growth of such weakly related functions, for instance the mutually dependent values of d, q and σ, we can fix a sufficiently large minimum value for σ, such that attacks that explore small standard deviations are mitigated [7]. Then we can focus on the choice of parameters d and q for a fixed standard deviation, for example $\sigma \approx 4$, which constitutes a good starting point for homomorphic encryption and, as we are going to see later, q just needs to be chosen big enough to accommodate the noise growth along the computation.

The Magma code in Table 3.2 implements a simplified version of the attack and can be used to study the security of LWE-based cryptography for small instances.

3.2.4 Dimension reduction

The decryption algorithm described in Definition 3.2.2 is similar when compared to ElGamal cryptosystem, because the ciphertext is formed by two polynomials, $c = [c_0, c_1]$, while the private key is given by $s = [1, s^\star]$. Hence, decryption can be represented by

$$m = [c_0 + c_1 s^\star]_q \pmod 2.$$

If we interpret s^* symbolically, the expression $c_0 + c_1 s^\star$ represents a polynomial of degree 1. To multiply two ciphertexts, $c = \text{ENC}_{pk}(m)$ and $c' = \text{ENC}_{pk}(m') = c'_0 + c'_1 s^\star$, we can compute

$$(c_0 + c_1 s^\star)(c'_0 + c'_1 s^\star) = c_0 c'_0 + (c_0 c'_1 + c'_0 c_1) s^\star + c_1 c'_1 (s^\star)^2.$$

If q is sufficiently big, as we replace $s^\star$ by the private key in last expression, we obtain a polynomial that can be used to recover $m.m'$. However, multiplication gives us back an element in higher dimension and since a compact cryptosystem cannot have such an increase in the ciphertext size, we must provide an algorithm to reduce the dimension of the ciphertext. This task is achieved by using an algorithm called SwitchKey, based on public parameters, that returns a ciphertext that can usually be decrypted.

Definition 3.2.3. Given a polynomial x, we define the algorithm BitDecomp, that returns $\log q$ binary polynomials x_i, where each coefficient of x is writen in binary representation, and the x_i coefficients are the corresponding i-th bits of this representation. Namely, we have that

$$\text{BitDecomp}(x) = [x_0, \ldots, x_{\log q}],$$

such that

$$x = \sum 2^i x_i.$$

Furthermore, consider the algorithm PowerOf, that returns $\log q$ polynomials in the form $2^i x$, as follows:

$$\text{PowerOf}(x) = [x, 2x, \ldots, 2^{\lfloor \log q \rfloor} x].$$

By construction, we have that

$$\langle \text{BitDecomp}(c), \text{PowerOf}(s) \rangle = \langle c, s \rangle \pmod q.$$

Definition 3.2.4. We define algorithm $\mathrm{SwitchKeyGen}$ as follows:

1. given a vector of polynomials s'', derived from private key sk, generate a random matrix $\overline{A'}$ with $\overline{N}$ rows and 2 columns, where $\overline{N} = 3\log^2 q$, compute $\overline{A} = \overline{A'}s'' + e'$, where e' is a vector of $\overline{N}$ rows whose elements are generated using distribution $\mathcal{D}$;
2. return $\overline{B} = \overline{A} + \mathrm{PowerOf}(s'')$, where $\mathrm{PowerOf}(s'')$ is added to the first column of $\overline{B}$.

Thus, given an expanded ciphertext $\bar{c}$, the algorithm $\mathrm{SwitchKey}$ can simply be defined as

$$\mathrm{SwitchKey}(\bar{c}) = \mathrm{BitDecomp}(\bar{c})^T\overline{B}.$$

Matrix $\overline{B}$ works as an alternative to the SSSP problem, described in previous constructions. In other words, it is analogous to the encryption of the private key using its own public key, such that we are again assuming circular security. Brakerski and Vaikuntanathan [24] proposed an alternative to make circular assumption unnecessary. They transformed the basic scheme in order to show that ciphertexts encryptying functions of the secret key are indistinguishable from ciphertexts encrypting zero.

3.2.5 Modulus reduction

Cryptosystems defined before BGV have a common problem: the noise grows quadratically with multiplications. In order to overcome this barrier, Brakerski and Vaikuntanathan [23] proposed a new technique to manage the noise. Basically, if the initial noise is proportional to r, after k levels of multiplications this noise would be proportional to r^{2^k}. The proposed solution was to use a decreasing moduli chain $q_i \approx q/r^i$. After the first multiplication, we adjust the ciphertext c, multiplying it by $1/r$ and fixing parity if necessary, and replacing modulus q by q/r. This change seems to bring no gain and you cannot repeat this procedure arbitrarily, because the chain decreases quickly (linearly with respect to the circuit depth) to a minimum value. However, it is easy to show that the noise is reduced in the same proportion $1/r$, that is, after the k-th multiplication, we obtain a noise proportional to r^k, instead of r^{2^k}. Therefore, there is an exponential gain involved in this transformation. When this chain reaches its end, it is necessary to use bootstrapping to continue the computation, but this subject is outside the scope of this work.

Definition 3.2.5. Given a vector of polynomials x, the algorithm $\mathrm{Scale}(x, q_i, q_{i+1})$ computes the vector of polynomials x' closest to $(q_{i+1}/q_i)x$, such that

$$x' = \mathrm{Scale}(x, q_i, q_{i+1}) = \lfloor (q_{i+1}/q_i)x \rceil \equiv x \pmod{t}.$$

Brakerski proposed a construction that avoids the scaling technique, called ***scale invariant*** [19]. In this construction, ciphertexts are composed by elements in the interval $(-1/2, 1/2]$, i.e., they are maintained in fractional form, with no need for modulus switching. Moreover, the construction have a classical reduction to lattice hard problems without using an exponential (in the degree d) modulus q. This reduction works only for the standard LWE and the scheme presents better performance only for $L \geq 20$. Therefore, determining if this technique is preferable will depend on the parameters. For some choices it will, for other choices the modulus switching approach will be the best option.

3.2.6 BGV

In this section we describe the BGV scheme [20], that can homomorphically deal with circuits of multiplicative depth at most L. Hence, if we know the maximum L necessary for a certain application, then we can derive optimal parameters for the use of the BGV scheme. Moreover, we avoid the expensive bootstrapping method.

Definition 3.2.6. **Setup.** Given security parameter λ and multiplicative depth L, compute $\mu = \theta(\log \lambda + \log L)$. For i varying from L to 0, run the $\textsc{Setup}(\lambda, (i+1)\mu)$ algorithm of scheme $\mathcal{E}_{R,i}$, obtaining a decreasing chain of moduli q_i.

Key Generation. Run $(sk_i, pk_i) = \mathcal{E}_{R,i}.\textsc{KeyGen}(\lambda)$ for each level i of the circuit. Compute $s'_i = sk_i \otimes sk_i$ and $s''_i = \mathrm{BitDecomp}(s'_i, q_i)$. Finally, compute $\overline{B_i} = \mathrm{SwitchKeyGen}(s''_i, sk_{i-1})$, for $i > 0$. Output the key pair (sk, pk), where the private key sk is formed by the values of sk_i, while public key pk corresponds to the public keys pk_i together with $\overline{B_i}$.

Encryption. Given the message $m \in \{0,1\}^t$, return $c = \textsc{Enc}_{pk_L}(m)$.

Decryption. Given the ciphertext c, at level i of the circuit, use the private key sk_i to compute $m = \textsc{Dec}_{sk_i}(c)$.

Consider the ciphertexts $c = c_0 + c_1 s^\star$ and $c' = c'_0 + c'_1 s^\star$. We have that addition $c + c'$ can be computed component-wise by

$$c + c' = (c_0 + c'_0) + (c_1 + c'_1)s^\star,$$

while multiplication is done by the tensor product of the ciphertexts, obtaining

$$c \times c' = c_0 c'_0 + (c_0 c'_1 + c'_0 c_1)s^\star + c_1 c'_1 (s^\star)^2,$$

where each of the coefficients is a polynomial and the vector composed by these three coefficients is called ***expanded ciphertext***. The algorithm $\mathrm{Recrypt}$ maps expanded ciphertexts to regular ciphertexts and is defined in Algorithm 3.1.

Algorithm 3.1 $\mathrm{Recrypt}$

INPUT The expanded ciphertext $\overline{c}$, the moduli q_i and q_{i+1}.
OUTPUT The ciphertext c.
$c = \mathrm{PowerOf}(\overline{c}, q_i)$ (the following condition is valid: $\langle c_1, s_i'' \rangle = \langle \overline{c}, s_i' \rangle$).
$c = \mathrm{Scale}(c, q_{i+1}, q_i)$.
$c = \mathrm{SwitchKey}(c, q_i, \overline{B_i})$.
return c.

Batch operations

In this section we will describe an important optimization to the scheme previously presented. The idea consists in using the Chinese Remainder Theorem (CRT) to allow simultaneous operations over a vector of messages. In the literature, this concept is associated with the SIMD model, because we have the capacity of parallel computation over vectors [94]. This parallelism allows to encode more information inside each ciphertext and therefore can be used to reduce the overhead of homomorphic encryption schemes. Concretely, it is possible to reduce the computation complexity per operation to a polylog overhead, achieving a big improvement with respect to the previous constructions. However, the circuit that will be homomorphically evaluated must have average width in $\Omega(\lambda)$ [46].

Furthermore, the capacity to perform additions and multiplications over vectors is not a complete computational model. For instance, it lacks the ability to permute vector elements. To solve this problem it is possible to use the Frobenius automorphisms over the ciphertext, obtaining circular rotations of the underlying vector elements. However, it is not possible to calculate every permutation using only circular rotations. In order to get any permutation, a permutation network is used, that allows us to combine left and right rotations to achieve more complicated permutations. More details can be found in the work of Halevi and Shoup [53].

Consider the integers n (usually a power of 2) and p, such that $p^d \equiv 1 \pmod{m}$. Thus $\mathbb{Z}_p$ contain an n-th primitive root of unity $\zeta_n \in \mathbb{Z}_p$, then the n-th cyclotomic polynomial $\phi_n(x)$ is such that its degree is equal to Euler's totient function $\varphi(n)$; furthermore it can be factored into $\ell = \varphi(n)/d$ degree-d terms modulo p

$$\phi_n(x) = \prod_{i \in \mathbb{Z}_m^\star} (x - \zeta_n^i) \pmod{p}.$$

A polynomial $a(x) \in \mathbb{Z}_p[x]/\Phi_n(x)$ can be represented by a vector containing the coefficients of the polynomial for each power of x, or it can be represented by a vector containing the evaluations of $a(x)$ over the n-th primitive roots of unity. Thus, there

are two possible representations: ***by coefficient*** or ***by evaluation***. The later is denoted by $\text{COEFS}(a(x))$, while the second one is denoted by $\text{EVALS}(a(x))$ and both representations are related by the Vandermonde matrix V_d as follows

$$\text{EVALS}(a(x)) = V_d\text{COEFS}(a(x)).$$

If the LWE problem modulus q itself is a product of primes $p_i \equiv 1 \pmod{n}$ (and $p_i \equiv 1 \pmod{p}$ to obtain better noise growth), we can use the so-called ***double CRT***, because the elements of $\mathbb{Z}_q$ can be decomposed with respect to the factors p_i and the factors of $\phi_n(x)$.

Recent use of cyclotomic rings is concentrated to the case where n is a power-of-two. As argued by Lyubashevsky, Peikert and Regev [69], this choice for n has both advantages and disadvantages. For instance, such a choice allows us to easily adapt the classical FFT in order to go from one representation to another, allowing faster arithmetic over the evaluation representation, because the operations can be computed component-wise in linear time. In general, choosing a power of two makes things simpler to understand and implement, which explains why many constructions adopted this choice. On the other hand, it would be better to choose n as the minimum integer satisfying Equation 3.2, while the next power-of-two may be twice as big as the best possible value, which constitutes an argument against such possibility. Nevertheless, this choice of n leads to larger ***expansion factors***. A work that shows many contributions related to the ring-LWE for non-power-of-two cyclotomic rings was proposed by Gentry, Halevi, Peikert and Smart [45].

3.2.7 Setting parameters

To provide parameters for the BGV cryptosystem, we must guarantee that the LWE problem is hard for all the moduli q_i, for $0 \leq i < L$. Also, we must guarantee that the ciphertext is decryptable for every i. These two conditions have a circular dependency, but although it is not simple to satisfy all the requirements, we will show how to find parameters that respect both of them. As previously defined, we must choose q_i with $(i+1)\mu$ bits, for $\mu = \theta(\log L + \log \lambda)$, but the hidden constant in this expression may assume different values depending on the SHE variant we are using.

If one previously determines the circuit to be evaluated, then one can compute the minimum q_{L-1} and the ratio between subsequent values q_i and q_{i-1} in order to allow the correctness of the homomorphic computation. Ana Costache and Nigel Smart [31] studied the parameter choice of many variants of the BGV and NTRU schemes, compared them, concluding that NTRU is better only for very small plaintext size. For $t > 5$, we have that BGV is the better choice. They considered two algorithms for SwitchKey and the possibility of using the scale invariant scheme. The noise growth depends on the initial noise and the number of additions and multiplications in the circuit. The authors constructed tables proposing parameters for the cases

$t \in \{2, 101, 2^{32}, 2^{64}, 2^{128}, 2^{256}\}$ and $L \in \{2, 5, 10, 20, 30\}$. They also generalized the definitions of $\mathrm{PowerOf}$ and $\mathrm{BitDecomp}$ to allow not only base $w = 2$, but any power of 2, turning it possible to look for optimal values for w. Choosing parameters is not an easy task, because we not only have the mentioned circular dependency, but also we have many details to determine depending on the desired application, as, for example the utilization of batch operations. Nevertheless, we can describe an abstract recipe that helps to accomplish this task:

1. establish a value for the standard deviation σ that avoids attacks described in the literature [7]. It is important to remark that this choice may or may not consider the worst-case connection to lattice problems, since we would need to obey the relation $\sigma = \omega(\sqrt{n})$ in order to have worst case reduction;

2. compute the minimal modulus q_{L-1} such that the scheme is correct for the highest level. This step depends on the base w in algorithms $\mathrm{PowerOf}$ and $\mathrm{BitDecomp}$. In special, it is possible to iterate through different values of w in order to find the optimal one;

3. compute the dimension n using Equation 3.2 to obtain the security of the cryptosystem;

4. calculate the size of the intermediate moduli q_i, for $0 \leq i < L$ and verify if the scheme is correct for all i.

Actually, we could invert our strategy, computing the minimal dimension n to obtain the correctness, and afterwards determine the value of q_L that provides the security according to Equation 3.2. If we have a special interest in some specific value for the modulus q_L, then it is possible to fix this parameter and recalculate the others in order to get correctness and security together.

A good measure for homomorphic encryption perfomance is the comparison of ciphertext size and plaintext size. However, it is important to remark that many homomorphic encryption schemes allow parallel computation of ℓ slots per encryted message. Table 3.3 shows the ciphertext size and number of slots of the BGV scheme for different values of L. The scale invariant scheme has smaller ciphertext size only for big L and, for $L \leq 20$, modulus switching technique has smaller ciphertext size. However, bigger ciphertext can encode bigger plaintexts, and depending on the target application it could be better to choose a determined setting or another.

3.3 Relations between the two problems

Recently, Cheon and Stehlé [29] showed that AGCD and LWE can be reduced to each other for a specific choice of parameters. This result is interesting because it helps to understand the LWE complexity by relating it to a simpler problem. Shortly, the LWE

	Modulus Switching		Scale Invariant	
L	size (KBits)	slots	size (KBits)	slots
2	424	1982	792	2714
5	2488	4817	3088	8567
10	10000	9664	10192	18170
20	40832	19542	37472	37742
30	91128	29199	82448	57130

Table 3.3: Ciphertext size and number of slots

problem in dimension n and modulo q can be transformed into an equivalent LWE problem in dimension 1 and modulo q^n and with much larger errors of size roughly equal to q^{n-1}. This problem can then be transformed into the problem of finding s in the torus $\mathbb{T}_{q^n}$, given samples of the form $(a + e)/s$, for uniformly distributed $a \in \mathbb{T}_{q^n}$ and small error e. This problem is called ***zero dimension LWE*** and it can be transformed into the AGCD problem. It is also possible to show that the AGCD can be transformed backwards, step-by-step, to the LWE problem. In the same paper, the authors proposed a SHE scheme to encrypt bits under the assumption that this modified AGCD problem is hard. This result allows to choose smaller η and, consequently, since $\gamma = \tilde{\Omega}(\eta^2)$, to choose much smaller γ, which corresponds in practice to the ciphertext size. We remark that it is an interesting open problem to extend this construction to be able to encrypt more than one bit, maintaining the validity of the reduction to the LWE problem.

Another interesting issue that appears in both AGCD and LWE problems is the utilization of a noise-free term, which in the case of the LWE problem is called ***first-is errorless LWE*** [22]. This condition seems to be an important tool for security reductions involving both the AGCD and the LWE problems. Although the noise-free term in principle could lead to worse security, there is no attack that can take advantage of this condition, while on the other hand it allows to prove important results. Hence it is important to understand what the role of such a condition in both problems is, pointing out what are the advantages and disadvantages of using it in practice. Considering the AGCD-based construction, it is crucial to use it to allow batch operations, thus the only way to build a secure CRT-based SHE scheme. However, in the LWE problem, it is used in the classical (instead of quantum) reduction of lattice hard problems to the intermediate LWE problem. Therefore, in the last case, the role of this condition is somewhat less important than in the first case. Nevertheless, this issue was not satisfactorily investigated in the literature and we leave it as a direction for further studies.

Chapter 4

Conclusion

> Progress is made by trial and failure; the failures are generally a hundred times more numerous than the successes; yet they are usually left unchronicled.
>
> William Ramsay

4.1 Final remarks

In this chapter, we summarize the results of our research. The main contribution of this work is an AGCD-based SHE scheme that is CCA1-secure and therefore avoids key-recovery attacks. The construction can be used to evaluate quadratic multivariate polynomials in such a way that the homomorphic computation can be ***verified***, i.e. that there is an algorithm that allows the receiver to verify if the cloud computed exactly the function that was asked. In this way, an adversary cannot submit queries that do not correspond to valid homomorphic evaluations of those functions, protecting the cryptosystem against malicious decryption queries. Such a cryptosystem is useful to evaluate many statistical functions over encrypted data, which is interesting, for example, to solve problems in the health and financial areas.

The restriction to quadratic multivariate polynomials in our verifiable computation scheme arises from the utilization of bilinear pairings. An interesting open problem, thus, is to construct a verifiable computation scheme under the same security model described in Section 2.4, but allowing more than one multiplication.

A cryptographic primitive that plays a central role in the scheme proposed in the previous chapter is the collision-resistant homomorphic hash function. We remark that such a primitive is interesting in its own, since it could hypothetically be used to construct CCA1-secure SHE schemes under the random oracle model, or it could be used in other interesting scenarios, since it is an important building block for other cryptographic primitives.

Homomorphic encryption is vulnerable to key-recovery attacks because, in general, the decryption algorithm is given by simple algebraic operations using the private key, such as multiplications and modular reductions. Hence, choosing appropriate ciphertexts to submit to a decryption oracle allows the easy obtention of information about the private key. We point out that, interestingly, the BGN cryptosystem is not vulnerable to key-recovery attacks, but it is not know to be CCA1-secure. However, the plaintext space of the scheme is small and this issue may be a problem for practical purposes. On the other hand, using SHE schemes it is possible to allow parallel computation over a vector of plaintext slots, which permits the encoding of more information inside each ciphertext, making it an interesting option when we want to evaluate the same function for many distinct input sets. In Section 2.4 we detailed the construction based on verifiable computation and compared our AGCD-based construction to the BGV-based proposal, obtaining a slightly better scheme.

Unfortunately, since we use as assumption the discrete logarithm and the factoring problems, the presented verifiable computation scheme is not protected against quantum adversaries. Thus the proposed encryption construction is not post-quantum.

Therefore, the conclusion of this thesis can be summarized by answering the following question.

> **How can we construct practical and useful SHE schemes that resists against key recovery attacks?**

We showed that under a restricted, but still useful, functionality, namely the restriction to compute only one level of multiplication of fresh ciphertexts, it is possible to obtain CCA1-security, avoiding key-recovery attacks. Statistical functions like average, variance and linear regression can be computed over encrypted data, allowing, for example, the use of private medical information of patients to compute correlations among a certain set of symptoms, trying to find better diagnostics. Another possible scenario is in the financial system, where statistical analysis is a key element in many cases. Our solution is based on the utilization of verifiable computation with homomorphic encryption to achieve a CCA1-secure scheme. The security of the scheme is based on the assumption that the AGCD problem is hard and we suggested a concrete instantiation of the parameters, obtaining interesting performance levels when compared to the BGV construction.

4.2 Future work

As future work, it is possible to explore other verifiable computation schemes, that are defined over a different security model and maybe can not be used to obtain a CCA1-secure SHE scheme, at least not in a straightforward manner. On the other hand, it still avoids key-recovery attacks. This would allow SHE schemes to perform

more than one multiplication, which is an important feature for the functionality of homomorphic encryption in cloud computing.

Bibliography

[1] GMP website. `https://gmplib.org/`. Accessed: 2016-05-18.

[2] NTL website. `http://www.shoup.net/ntl/`. Accessed: 2016-05-18.

[3] D. Aharonov and O. Regev. Lattice problems in NP ∩ coNP. In *In IWPEC, volume 5018 of Lecture Notes in Computer Science*, page 765. Springer, 2005.

[4] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, STOC '96, pages 99–108, New York, NY, USA, 1996. ACM.

[5] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, STOC '97, pages 284–293, New York, NY, USA, 1997. ACM.

[6] D. F. Aranha and C. P. L. Gouvêa. RELIC is an Efficient Library for Cryptography. `http://code.google.com/p/relic-toolkit/`.

[7] S. Arora and R. Ge. New algorithms for learning in presence of errors. In Luca Aceto, Monika Henzinger, and Jiri Sgall, editors, *ICALP (1)*, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415. Springer, 2011.

[8] L. Babai. On Lovász lattice reduction and the nearest lattice point problem. *Combinatorica*, 6, 1986.

[9] P. S. L. M. Barreto, F. P. Biasi, R. Dahab, J. C. López-Hernández, E. M. Morais, A. D. S. Oliveira, G. C. C. F. Pereira, and J. E. Ricardini. Introdução a criptografia pós-quântica. In Rossana Maria de Castro Andrade, editor, *Minicursos do XIII Simpósio em Segurança da Informação e de Sistemas Computacionais. 193ed.*, 2013.

[10] P. S. L. M. Barreto, F. P. Biasi, R. Dahab, J. C. López-Hernández, E. M. Morais, A. D. S. Oliveira, G. C. C. F. Pereira, and J. E. Ricardini. A panorama of post-quantum cryptography. In *Open Problems in Mathematics and Computational Science. 1ed.: Springer International Publishing*, pages 387–439, 2014.

[11] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, CCS '93, pages 62–73, New York, NY, USA, 1993. ACM.

[12] D. J. Bernstein, J. Buchmann, and E. Dahmen. *Post-Quantum Cryptography*. Springer, Heidelberg, Deutschland, 2008.

[13] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50:506–519, July 2003.

[14] D. Boneh, C. Gentry, and M. Hamburg. Space-efficient identity based encryption without pairings. In *FOCS*, pages 647–657, 2007.

[15] J. W. Bos, K. Lauter, J. Loftus, and M. Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In Martijn Stam, editor, *IMA Int. Conf.*, volume 8308 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2013.

[16] J. W. Bos, K. Lauter, and M. Naehrig. Private predictive analysis on encrypted medical data. *Journal of Biomedical Informatics*, 50:234–243, 2014.

[17] A. M. Braga and E. M. Morais. Implementation issues in the construction of standard and non-standard cryptography on android devices. In *Securware, The Eighth International Conference on Emerging Security Information, Systems and Technologies*, pages 144–150, 2014.

[18] A. M. Braga, E. M. Morais, D. C. Schwab, A. L. Vannucci, and R. Zanco Neto. Integrated technologies for communication security and secure deletion on android smartphones. *ICQMN*, 8:28, 2015.

[19] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In *Advances in Cryptology - Crypto 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886. Springer, 2012.

[20] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. Fully homomorphic encryption without bootstrapping. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:111, 2011.

[21] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 309–325, New York, NY, USA, 2012. ACM.

[22] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 575–584, New York, NY, USA, 2013. ACM.

[23] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, pages 97–106, 2011.

[24] Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *Proceedings of the 31st Annual Conference on Advances in Cryptology*, CRYPTO'11, pages 505–524, Berlin, Heidelberg, 2011. Springer-Verlag.

[25] J. Buchmann, D. Cabarcas, F. Göpfert, A. Hülsing, and P. Weiden. *Selected Areas in Cryptography – SAC 2013: 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers*, chapter Discrete Ziggurat: A Time-Memory Trade-Off for Sampling from a Gaussian Distribution over the Integers, pages 402–417. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[26] Y. Chen and P. Nguyen. Faster algorithms for approximate common divisors: Breaking fully-homomorphic-encryption challenges over the integers. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 502–519. Springer, 2012.

[27] Y. Chen and P. Q. Nguyen. Bkz 2.0: Better lattice security estimates. In D. H. Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011: 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, pages 1–20, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[28] J. Cheon, J. Coron, J. Kim, M. Lee, T. Lepoint, M. Tibouchi, and A. Yun. Batch fully homomorphic encryption over the integers. In T. Johansson and P. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 315–335. Springer Berlin Heidelberg, 2013.

[29] J. Cheon and D. Stehlé. Fully homomophic encryption over the integers revisited. In E. Oswald and Marc F., editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Computer Science*, pages 513–536. Springer Berlin Heidelberg, 2015.

[30] H. Cohn and N. Heninger. Approximate common divisors via lattices. Cryptology ePrint Archive, Report 2011/437, 2011. `http://eprint.iacr.org/`.

[31] A. Costache and N. P. Smart. Which ring based somewhat homomorphic encryption scheme is best? Cryptology ePrint Archive, Report 2015/889, 2015. `http://eprint.iacr.org/`.

[32] R. Dahab, S. Galbraith, and E. M. Morais. Adaptive key recovery attacks on NTRU-based somewhat homomorphic encryption schemes. In A. Lehmann and S. Wolf, editors, *Information Theoretic Security*, volume 9063 of *Lecture Notes in Computer Science*, pages 283–296. Springer International Publishing, 2015.

[33] R. Dahab and E. M. Morais. Encriptação homomórfica. In Santin A. (Org.) Maziero C. (Org.) Santos, A. L. (Org.) and P. A. S. Gonçalves, editors, *Minicursos do XII Simpósio em Segurança da Informação e de Sistemas Computacionais. 12ed.*, pages 1–195, 2012.

[34] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, (22), 1976.

[35] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. *Advances in Cryptology – CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, chapter Lattice Signatures and Bimodal Gaussians, pages 40–56. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[36] D. S. Dummit and R. M. Foote. *Abstract Algebra*. Wiley, 2003.

[37] N. C. Dwarakanath and S. D. Galbraith. Sampling from discrete gaussians for lattice-based cryptography on a constrained device. *Applicable Algebra in Engineering, Communication and Computing*, 25(3):159–180, 2014.

[38] Y. Elias, K. E. Lauter, E. Ozman, and K. E. Stange. Provably weak instances of ring-lwe. In R. Gennaro and M. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, pages 63–92, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[39] D. Fiore, R. Gennaro, and V. Pastro. Efficiently verifiable computation on encrypted data. Cryptology ePrint Archive, Report 2014/202, 2014. `http://eprint.iacr.org/`.

[40] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT*, pages 1–17, 2013.

[41] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *IACR Cryptology ePrint Archive*, 2013:451, 2013.

[42] C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`.

[43] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 169–178, New York, NY, USA, 2009. ACM.

[44] C. Gentry. Encrypted messages from the heights of cryptomania. In *TCC*, pages 120–121, 2013.

[45] C. Gentry, S. Halevi, C. Peikert, and N. P. Smart. Ring switching in BGV-style homomorphic encryption. Cryptology ePrint Archive, Report 2012/240, 2012. `http://eprint.iacr.org/`.

[46] C. Gentry, S. Halevi, and N. P. Smart. Fully homomorphic encryption with polylog overhead. In *EUROCRYPT*, pages 465–482, 2012.

[47] C. Gentry, S. Halevi, and N. P. Smart. Homomorphic evaluation of the AES circuit. *IACR Cryptology ePrint Archive*, 2012:99, 2012.

[48] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 197–206, New York, NY, USA, 2008. ACM.

[49] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pages 197–206, New York, NY, USA, 2008. ACM.

[50] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO (1)*, pages 75–92, 2013.

[51] O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology—CRYPTO '97*, Lecture Notes in Computer Science, pages 112–131. Springer-Verlag, 1997.

[52] K. Lauter H. Chen and K. E. Stange. Attacks on search rlwe. Cryptology ePrint Archive, Report 2015/971, 2015. `http://eprint.iacr.org/`.

[53] S. Halevi and V. Shoup. Algorithms in helib. Cryptology ePrint Archive, Report 2014/106, 2014. `http://eprint.iacr.org/`.

[54] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[55] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In *Lecture Notes in Computer Science*, pages 267–288. Springer-Verlag, 1998.

[56] N. Howgrave-Graham. Approximate integer common divisors. In *CaLC*, pages 51–66, 2001.

[57] S. Khot. Inapproximability results for computational problems on lattices, 2007. survey paper preparedfor the lll+25 conference. to appear. [35. In *In Proc. 11th Annual ACM-SIAMSymposium on Discrete Algorithms*, pages 937–941. Combinatorica, 2000.

[58] J. C. Lagarias. The computational complexity of simultaneous diophantine approximation problems. *SIAM J. Comput.*, 14(1):196–209, February 1985.

[59] K. Lauter, A. Lopez-Alt, and M. Naehrig. Private computation on encrypted genomic data. Technical Report MSR-TR-2014-93, Microsoft, June 2014.

[60] A. K. Lenstra. Factoring multivariate polynomials over algebraic number fields. *SIAM J. Comput*, (16), 1987.

[61] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[62] T. Lepoint. *Design and Implementation of Lattice-Based Cryptography*. PhD thesis, École Normale Supérieure and University of Luxembourg, June 2014.

[63] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 62–91. Springer Berlin Heidelberg, 2010.

[64] R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In *Proceedings of the 11th International Conference on Topics in Cryptology: CT-RSA 2011*, CT-RSA'11, pages 319–339, Berlin, Heidelberg, 2011. Springer-Verlag.

[65] J. Loftus, A. May, N. P. Smart, and F. Vercauteren. On CCA-secure somewhat homomorphic encryption. In A. Miri and S. Vaudenay, editors, *Selected Areas in Cryptography*, volume 7118 of *Lecture Notes in Computer Science*, pages 55–72. Springer Berlin Heidelberg, 2012.

[66] V. Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In M. Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009: 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, pages 598–616, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[67] V. Lyubashevsky. Lattice signatures without trapdoors. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012: 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 738–755, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[68] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. *Advances in Cryptology EUROCRYPT 2010*, 6110/2010(015848):1?23, 2010.

[69] V. Lyubashevsky, C. Peikert, and O. Regev. A toolkit for ring-LWE cryptography. In *EUROCRYPT*, pages 35–54, 2013.

[70] L. Ducas M. Albrecht, S. Bai. A subfield lattice attack on overstretched ntru assumptions: Cryptanalysis of some fhe and graded encoding schemes. Cryptology ePrint Archive, Report 2016/127, 2016. `http://eprint.iacr.org/`.

[71] D. Micciancio. On the hardness of the shortest vector problem. Technical report, 1998.

[72] D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *computational complexity*, 16(4):365–411, 2007.

[73] D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, March 2002.

[74] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer Berlin Heidelberg, 2012.

[75] D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, April 2007.

[76] E. M. Morais, D. F. Aranha, and R. Dahab. AGCD-based CCA1-secure somewhat homomorphic encryption using verifiable computation. [Submitted].

[77] E. M. Morais, D. F. Aranha, and R. Dahab. Homomorphic encryption. Technical Report 02, Institution of Computing, 2016. `http://www.ic.unicamp.br/~reltech/2016/16-02.pdf`.

[78] E. M. Morais, D. F. Aranha, and R. Dahab. Lattice-based cryptography. Technical Report 01, Institution of Computing, 2016. `http://www.ic.unicamp.br/~reltech/2016/16-01.pdf`.

[79] M. Naehrig, K. Lauter, and V. Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, CCSW '11, pages 113–124, New York, NY, USA, 2011. ACM.

[80] P. Nguyen and O. Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 271–288. Springer Berlin Heidelberg, 2006.

[81] P. Nguyen and J. Stern. The two faces of lattices in cryptology. In J. H. Silverman, editor, *Cryptography and Lattices: International Conference, CaLC 2001 Providence, RI, USA, March 29–30, 2001 Revised Papers*, pages 146–180, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/3-540-44670-2_12.

[82] P. Q. Nguyen and B. Valle. *The LLL Algorithm: Survey and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[83] T. Oder, T. Pöppelmann, and T. Güneysu. Beyond ecdsa and rsa: Lattice-based digital signatures on constrained devices. In *DAC*, pages 110:1–110:6. ACM, 2014.

[84] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 333–342, New York, NY, USA, 2009. ACM.

[85] C. Peikert. *Advances in Cryptology – CRYPTO 2010: 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, chapter An Efficient and Parallel Gaussian Sampler for Lattices, pages 80–97. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[86] C. Peikert. A decade of lattice cryptography. Cryptology ePrint Archive, Report 2015/939, 2015. http://eprint.iacr.org/.

[87] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 84–93, New York, NY, USA, 2005. ACM.

[88] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.

[89] O. Regev. The learning with errors problem (invited survey). In *IEEE Conference on Computational Complexity*, pages 191–204. IEEE Computer Society, 2010.

[90] R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation, Academia Press*, pages 169–179, 1978.

[91] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 26:96–99, January 1983.

[92] P. Rogaway. The moral character of cryptographic work. Cryptology ePrint Archive, Report 2015/1162, 2015. http://eprint.iacr.org/.

[93] A. Sahai and B. Waters. Attribute-based encryption for circuits from multilinear maps. *CoRR*, abs/1210.5287, 2012.

[94] N. P. Smart and F. Vercauteren. Fully homomorphic SIMD operations. *IACR Cryptology ePrint Archive*, 2011:133, 2011.

[95] D. Stehlé and R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In *Proceedings of the 30th Annual international conference on Theory and applications of cryptographic techniques: advances in cryptology*, EUROCRYPT'11, pages 27–47, Berlin, Heidelberg, 2011. Springer-Verlag.

[96] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *Proceedings of the 29th Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'10, pages 24–43, Berlin, Heidelberg, 2010. Springer-Verlag.

Index